GETTING AHEAD

Commodore 64™

SPACE ADVENTURES

for the Commodore 64

HAYES

SPACE ADVENTURES

for the Commodore 64

Programs Compiled and Edited by Peter Lear

Editor:	Cyril Hayes
Design:	Karen McArdle
	Margaret Horvath
Illustration:	Richard Livingston

Hayes Publishing Ltd., 3312 Mainway, Burlington, Ontario L7M 1A7

NOTE

The publisher and authors have made every effort to assure that the computer programs are accurate and complete. However, this publication is prepared for general readership, and neither the publisher nor the authors have any knowledge about or ability to control any third party's use of the programs and programming information. There is no warranty or representation by the publisher or the authors that the programs or programming information in this book will enable the reader or user to achieve any particular result.

Copyright©1984 by Hayes Publishing Ltd. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without permission from the publisher.

©1984 Hayes Publishing Ltd., Burlington, Ontario

ISBN 0-88625-066-8

Commodore 64 is a registered trademark of
Commodore Computer Systems.

CONTENTS

Introduction	4
MOONSHUTTLE	6
METEOR SHOWER	12
About DATA Statements	17
ALIEN ATTACK	18
About READ Statements	29
RED ALERT	30
SPACE GUARD	36
How A Game Is Made	48

INTRODUCTION

This book contains listings for games written in BASIC for the Commodore 64 computer made by Commodore Business Machines. All of the games will work on any Commodore 64 with no cartridges plugged in.

TYPING IN THE PROGRAMS

The program listings are set up like the listings in your Commodore 64 owner's manual. To make the keying of the programs easier, a standard has been used. Whenever you encounter brace brackets ({ }) in a program listing, refer to the following guide.

To do this:	Type this:		
[SC]	SHIFT	AND	CLR HOME
[CU]	SHIFT	AND	CRSR ↑ ↓
[CL]	SHIFT	AND	CRSR ← →
[IN]	SHIFT	AND	INST/ DEL
[RV]	CTRL	AND	) 9
[RD]	CTRL	AND	# 3
[RO]	CTRL	AND	0
[BL]	CTRL	AND	' 7

To do this:	Type this:		
[YL]	CTRL	AND	(8
[BK]	CTRL	AND	! 1
[PU]	CTRL	AND	% 5
[CY]	CTRL	AND	$ 4
[GR]	CTRL	AND	& 6
[WH]	CTRL	AND	" 2
[CR]	CRSR ← →		
[CD]	CRSR ↑ ↓		
[HM]	CLR HOME		

When typing in these programs you can save some time by using the Commodore abbreviations for some of the BASIC language words. You know from reading your owner's manual that the question mark (?) is a short way of typing PRINT. Most other BASIC words can be shortened by typing in the first letter normally, followed by the second letter together with the shift key. This will cause a graphic character to appear instead of the second letter. The C 64 will interpret this as an entire BASIC word. Here are some examples:

To Get	You Type	As It Appears On The Screen
RUN	R shifted U	R
LIST	L shifted I	L
GOTO	G shifted O	G

There are exceptions to this rule. The words RETURN, RESTORE, GOSUB, CLOSE, CLR, LET, STEP, and STR$ can be abbreviated by typing the first two letters normally and shifting the third. Here are some examples:

To Get	You Type	As It Appears On The Screen
GOSUB	GO shifted S	GO
RETURN	RE shifted T	RE

The words IF, ON, ST and TI cannot be shortened.

Accidents do happen, you could type in most of a program only to have a power failure and lose it all, or you could accidentally RUN a program with mistakes in it that destroy the program. So SAVE copies of the games on tape or floppy disk as you enter them. Do so often. Once you have a final copy of the game, you can record over the other versions on tape or delete the disk versions.

DEBUGGING

After typing in a program and SAVEing it, the C 64 may have trouble RUNning it. This sometimes happens when copying a program from paper to computer. Check that you have entered all the program lines correctly and completely.

A common problem is the famous SYNTAX ERROR. If you get one, LIST the program line on the screen. Compare it to the book listing. You will probably see one of these problems:

1. Spelling error, or
2. Punctuation error (brackets, commas, colons or semi-colons missing), or
3. The number zero (Ø) confused with the letter 'O' (or vice versa), or
4. The number one (1) confused with the letter 'I' (also vice versa).

Fix the error and RUN the program again. It may take several attempts to get all the bugs out, but the work will be worth it. Remember to SAVE a final copy that has all the corrections made.

EXPERIMENT

If you have some programming knowledge, do not be afraid to try changing the games. After each program description there is a list of the program's variables and how they are used. And remember, there is nothing that you can enter into the C 64 by program or through the keyboard that can do any permanent damage. If something goes wrong, you can turn the C 64 off and on to begin again.

MOON SHUTTLE

adapted by Peter J. Lear
(originally for the VIC 20 by P.A. Roberts)

Five...four...three...two...one...lift off!

You have just left Phobos, one of the two moons of the planet Mars, and now you must land on Deimos, the other moon. But landing here isn't an easy task. You must navigate your moonshuttle through a deadly asteroid field. Using your four thrusters, U to go up, J to move right, H to move left and N to go down, you can maneuver through your descent and avoid the asteroids. To keep track of your fuel and energy consumption, read-outs of this information are given periodically.

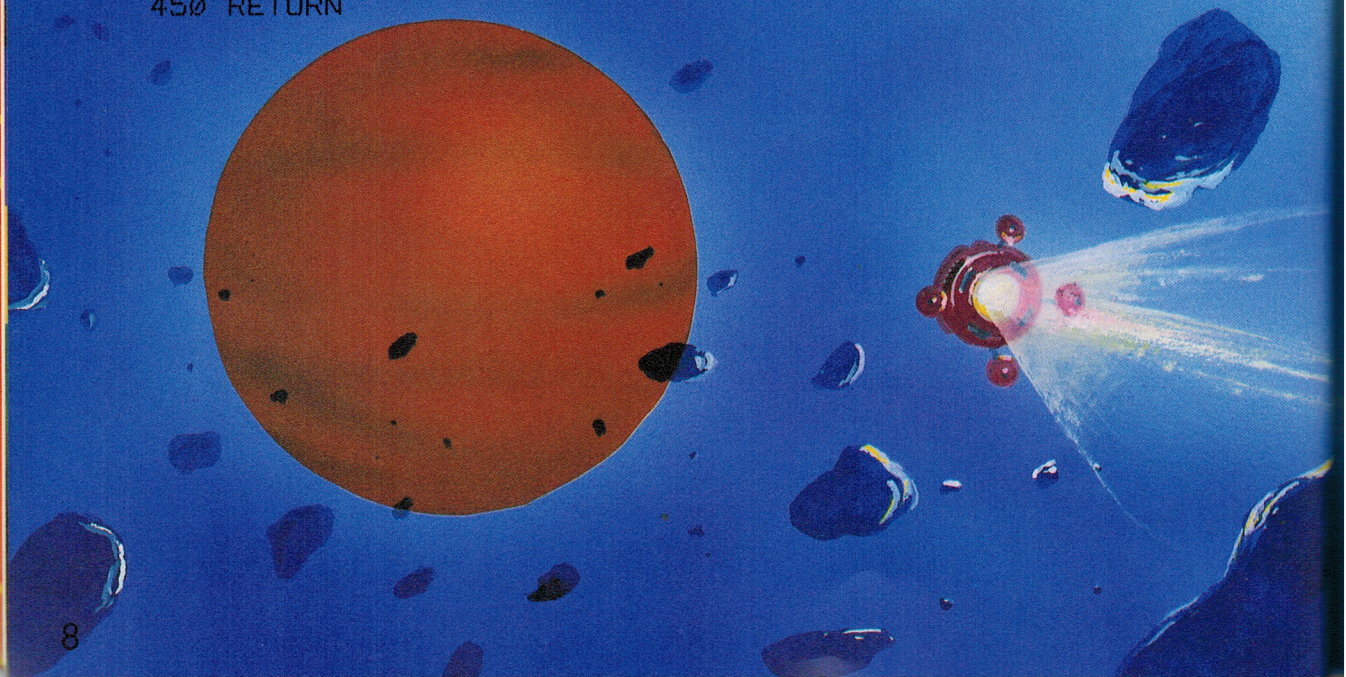
VARIABLES USED:

MAIN PROGRAM:

D, DI	– difficulty factor
A, B	– moonshuttle
Z	– E (energy) meter
H	– height
V	– fuel
U, Q	– screen
P, U	– moon base
VC	– volume
VN	– noise
J	– keyboard
T	– delay counter
K	– number of asteroids
W	– red alert signal
Y	– game over flag
L, M	– sound counter
RE$	– keyboard response
I	– counter
VH,VL	– sound frequency

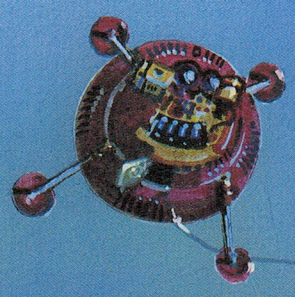
MOONSHUTTLE PROGRAM

```
100 PRINT"{SC}{CD}{CD}":POKE53280,4:POKE53281,7
110 PRINTTAB(16);"{RD}MOONSHUTTLE"
120 PRINTTAB(15);"{WH}{CD}BY PA ROBERTS"
130 PRINTTAB(10);"ADAPTED BY PETER J LEAR"
140 PRINT"{BK}{CD}  YOUR JOB IS TO LAND THE MOON SHUTTLE"
150 PRINTTAB(4);"ON ONE OF THE BLUE LANDING PADS"
160 PRINTTAB(15);"{BL}{CD}USE THE  KEYS:{CD}"
170 PRINTTAB(20);"UP"
180 PRINTTAB(20);"U"
190 PRINTTAB(14);"LEFT H J RIGHT"
200 PRINTTAB(20);"N"
210 PRINTTAB(18);"DOWN"
220 PRINTTAB(16);"{GR}{CD}PLEASE WAIT"
230 DATA 24,60,126,255,126,90,90,231
240 DATA 255,255,255,255,255,255,255,255
250 DATA 90,165,90,165,90,60,24,24
260 DATA 60,126,189,90,165,90,165,66
270 DATA 10,64,17,129,62,221,231,182
280 POKE52,48:POKE51,0:POKE56,48:POKE55,0
290 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
300 FORI=0TO2047:POKE12288+I,PEEK(53248+I):NEXTI
310 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
320 FORI=12288TO12327:READJ:POKEI,J:NEXTI
330 PRINTTAB(16);"{CD}PRESS A KEY{BK}"
340 POKE198,0
350 GETRE$:IFRE$=""THEN350
360 FORI=54272TO54296:POKEI,0:NEXTI
370 POKE54277,128:POKE54278,128
380 POKE54284,128:POKE54285,128
390 PRINT"{SC}{CD}{CD}DIFFICULTY (0-5)":INPUTD
400 IFD<0ORD>5THEN390
410 PRINT"{SC}"
420 DI=(D*20+10):POKE53272,(PEEK(53272)AND240)+12:GOTO580
430 IF PEEK(A)<>230ANDPEEK(A)<>32ANDPEEK(A)<>232THENW=2
440 IF PEEK(A)=46 THENPRINT"{HM}{PU}SHIP HIT BADLY          "
450 RETURN
```

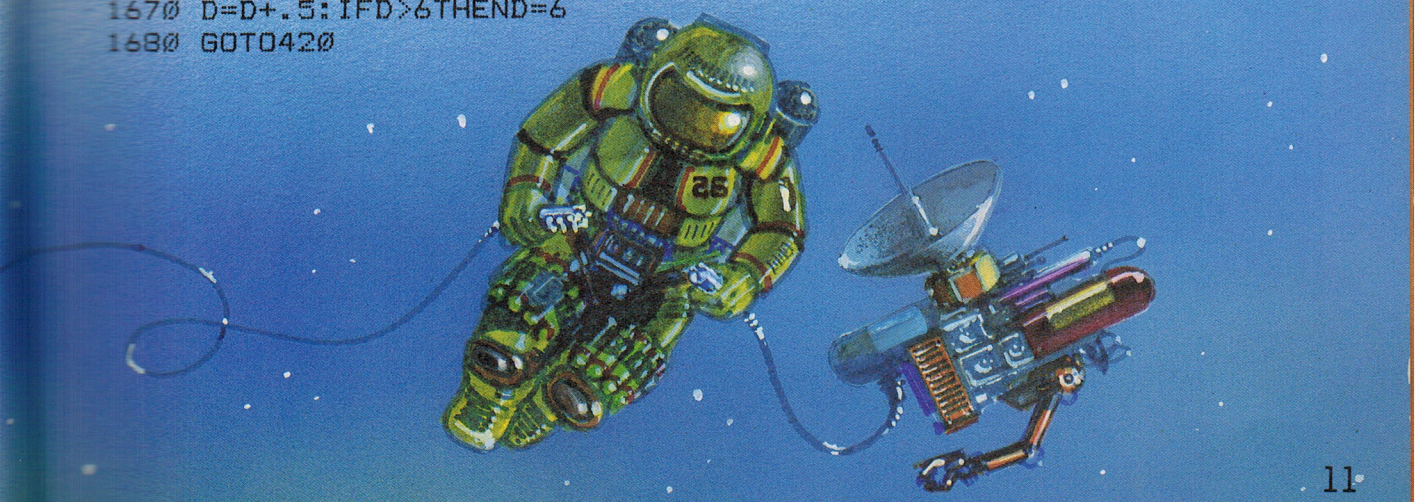


```
460 IFA=1868ORA=1867THENY=2:E=E+10
470 IFA=1888ORA=1887THENY=2:E=E+10
480 IFA=1962THENY=2:E=E+60
490 IFA=1874ORA=1874THENY=2:E=E+30
500 IFA=1878ORA=1878THENY=2:E=E+20
510 RETURN
520 Z=Z-1:IFZ<=0THENPRINT"{HM}{PU}OUT OF FUEL!          ":V=2
530 RETURN
540 IFA<1344ANDV<>2THENPRINT"{HM}{CY}H=";H;"E=";Z
550 IFA>=1704ANDV<>2THEN PRINT"{HM}{CY}H=";H/10;" {PU}{RV}E-
METER FAULTY!{RV}      ":RETURN
560 IFA>=1344ANDV<>2THENPRINT"{HM}{CY}H=";(H/10);"E/10=";(Z/
10)
570 RETURN
580 POKE53280,0:POKE53281,0:Q=56176:P=1904:FORW=QTO56295:POK
EW,1:NEXTW
590 POKEP,111:POKEP+7,111:POKEP+1,100
600 POKEP+20,111:POKEP+27,111:POKEP+21,100
610 POKEP+8,100:POKEP+12,100:POKEP+2,121:POKEP+6,121
620 POKEP+28,100:POKEP+32,100:POKEP+22,121:POKEP+26,121
630 POKEP+15,121:POKEP+5,98:POKEP+13,98
640 POKEP+35,121:POKEP+25,98:POKEP+33,98
650 FORU=1944TO2023:POKEU,1:NEXTU
660 POKEP+9,1:POKEP+11,1
670 POKEP+29,1:POKEP+31,1
680 POKEQ+3,3:POKEQ+4,3:POKEQ+10,3:POKEQ+14,3
690 POKEP+3,104:POKEP+4,104:POKEP+10,104:POKEP+14,104
700 POKEQ+23,3:POKEQ+24,3:POKEQ+30,3:POKEQ+34,3
710 POKEP+23,104:POKEP+24,104:POKEP+30,104:POKEP+34,104
720 POKE2002,102:POKE56274,3
730 PRINT"{HM}{RV}{WH}PRESS A KEY"
740 POKE56234,3:POKE1962,0:IFPEEK(197)=64THEN740
750 PRINT"{HM}                    "
760 FORU=300TO0STEP-1:PRINT"{HM}{CY}";INT(U/50):NEXTU
770 U=56234:P=1962:VN=54276:VC=54296:J=197:H=210:W=0:Y=0:Z=4
1:V=0
780 VH=54273:VL=54272
790 POKEVC,15:POKEVN,129:POKEVH,20:POKEVL,30
800 POKEP,3:POKEU,2
810 P=P-40:U=U-40
820 POKEP,0:POKEU,3:FORT=1TO50:NEXTT
830 IFP=1082THEN860
840 POKEP,32:POKEP+40,32
850 GOTO800
860 POKEVN,0:POKEVC,0:U=U+40:P=P+40:POKEP,32
870 FORK=1TODI:M=INT((RND(1)*500)+1):POKE1104+M,46:POKE55376
+M,INT(RND(1)*6+1):NEXTK
880 A=1082:B=55354:W=0
```

```
890 IFV=2THENPRINT"{HM}{PU}OUT OF FUEL!              "
900 GOSUB540
910 IFPEEK(J)=30THEN1000
920 IFPEEK(J)=34THEN1100
930 IFPEEK(J)=29THEN1220
940 IFPEEK(J)=39THEN1320
950 POKEVC,0
960 IFA>=1424THENPOKEA,32:A=A+1:B=B+1:GOSUB430:POKEA,0:POKEB
,3
970 IFA<1424THENPOKEA,32:A=A-1:B=B-1:GOSUB430:POKEA,0:POKEB,
3
980 IFW=2THEN1440
990 FORT=1TO40:NEXTT:GOTO890
1000 POKEVC,15:POKEVN,129
1010 GOSUB520:IFV=2THEN950
1020 GOSUB1170:A=A-40:B=B-40
1030 IFA<1064THENA=1064:B=55336
1040 GOSUB430:GOSUB460
1050 POKEA,0:POKEB,3:H=H+10:GOSUB540
1060 IFW=2THEN1440
1070 IFY=2THEN1620
1080 IFPEEK(J)<>51THENPOKEVC,0:POKEVN,0:GOTO910
1090 GOTO1010
1100 POKEVC,15:POKEVN,129
1110 GOSUB520:IFV=2THENGOTO950
1120 GOSUB1170:A=A+41:B=B+41
1130 IFA<1064THENA=1064:B=55336
1140 GOSUB430:GOSUB460
1150 POKEA,0:POKEB,3:H=H-10:GOSUB540
1160 GOTO1180
1170 POKEA+40,3:POKEB+40,2:FORT=1TO90:NEXTT:POKEA,32:POKEA+4
0,32:RETURN
1180 IFW=2THEN1440
1190 IFY=2THEN1620
1200 IFPEEK(J)<>20THENPOKEVC,0:POKEVN,0:GOTO910
1210 GOTO1110
1220 POKEVC,15:POKEVN,129
1230 GOSUB520:IFV=2THEN950
1240 GOSUB1170:A=A+39:B=B+39
1250 IFA<1064THENA=1064:B=55336
1260 GOSUB430:GOSUB460
1270 POKEA,0:POKEB,3:H=H-10:GOSUB540
1280 IFW=2THEN1440
```

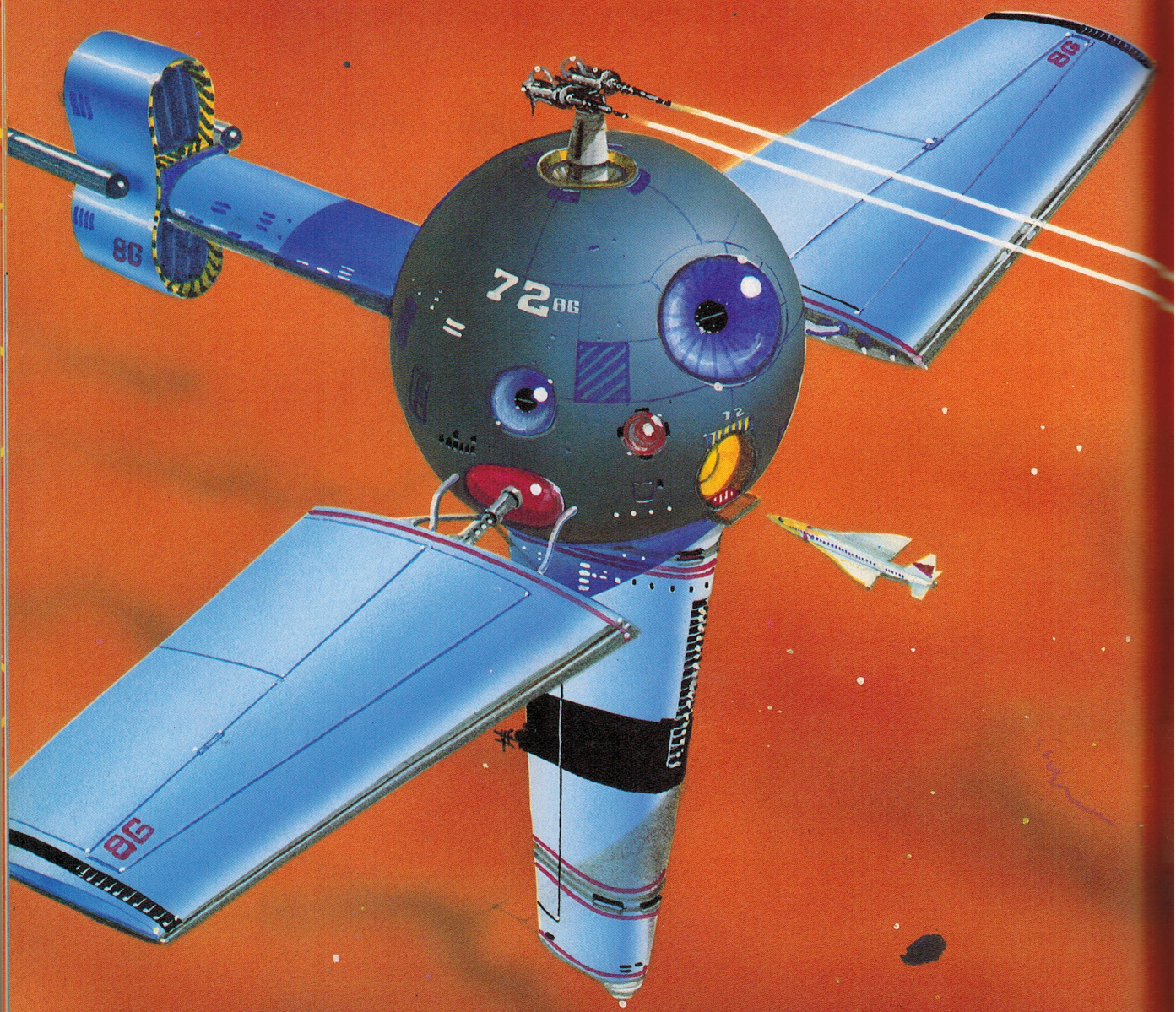


```
1290 IFY=2THEN1620
1300 IFPEEK(J)<>43THENPOKEVC,0:POKEVN,0:GOTO910
1310 GOTO1230
1320 POKEVC,15:POKEVN,129
1330 GOSUB520:IFV=2THEN950
1340 A=A+40:B=B+40
1350 GOSUB430:GOSUB460
1360 A=A-40:B=B-40
1370 GOSUB1170:A=A+40:B=B+40
1380 IFA<1064THENA=1064:B=55336
1390 POKEA,0:POKEB,3:H=H-10:GOSUB540
1400 IFW=2THEN1440
1410 IFY=2THEN1620
1420 IFPEEK(J)<>28THENPOKEVC,0:POKEVN,0:GOTO910
1430 GOTO1330
1440 POKEVN,129:POKEVH,15:POKEVL,20:POKEVC,15:PRINT"{HM}{PU}
{RV}RED ALERT{RO}                    "
1450 U=INT((1944-A)/40)
1460 POKEVC,15
1470 FORL=UTO0STEP-1:POKEA,32:A=A+40:B=B+40:POKEA,0:POKEB,IN
T(RND(1)*5+3)
1480 FORT=1TO8:NEXTT:NEXTL
1490 POKE54283,33:POKE54280,22
1500 FORL=0TO6
1510 FORM=0TO255STEP12:POKE54279,M
1520 POKEA,0:POKEB,INT(RND(1)*5+3)
1530 NEXTM
1540 FORI=1TO20:NEXTI
1550 NEXTL
1560 POKEVN,0:POKE54283,0
1570 POKEVN,129:POKEVH,20:POKEVL,30:FORL=15TO0STEP-.05
1580 POKEA,2:POKEB,INT(RND(1)*4+3)
1590 POKEVC,L:NEXTL:POKEVN,0:POKEVC,0
1600 POKEA,4:POKEB,3
1610 FORT=1TO3000:NEXT:PRINT"{SC}":GOTO420
1620 PRINT"{HM}{GR}{RV}SAFE LANDING!{RO}                    "
1630 POKEVC,0:POKEVN,0
1640 FORT=1TO1000:NEXT
1650 PRINT"{SC}{PU}{RV}SCORE={RO}";E*DI;"   "
1660 FORT=1TO1000:NEXT
1670 D=D+.5:IFD>6THEND=6
1680 GOTO420
```



METEOR SHOWER

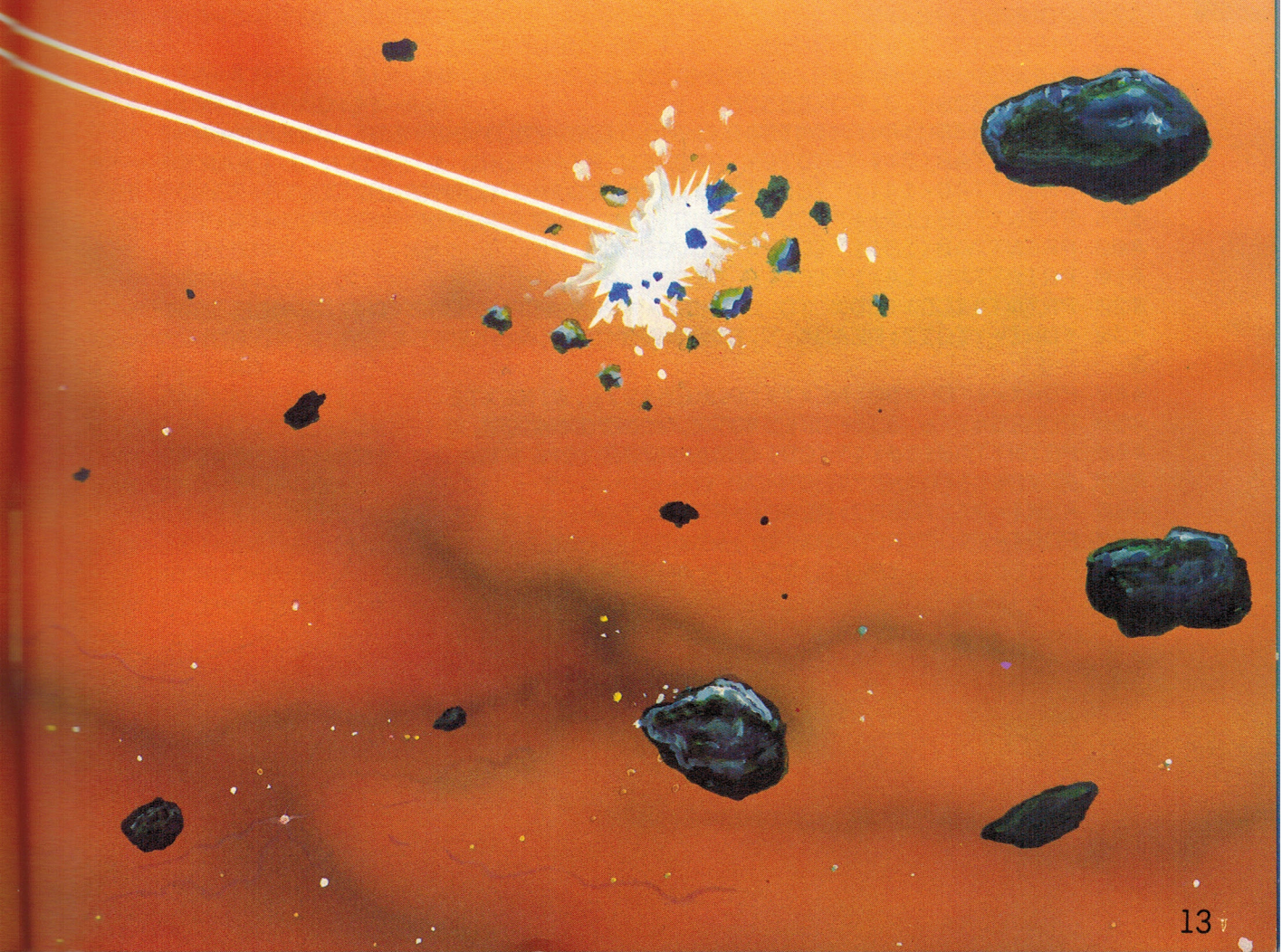
by Andrew Stanford and Peter Lear



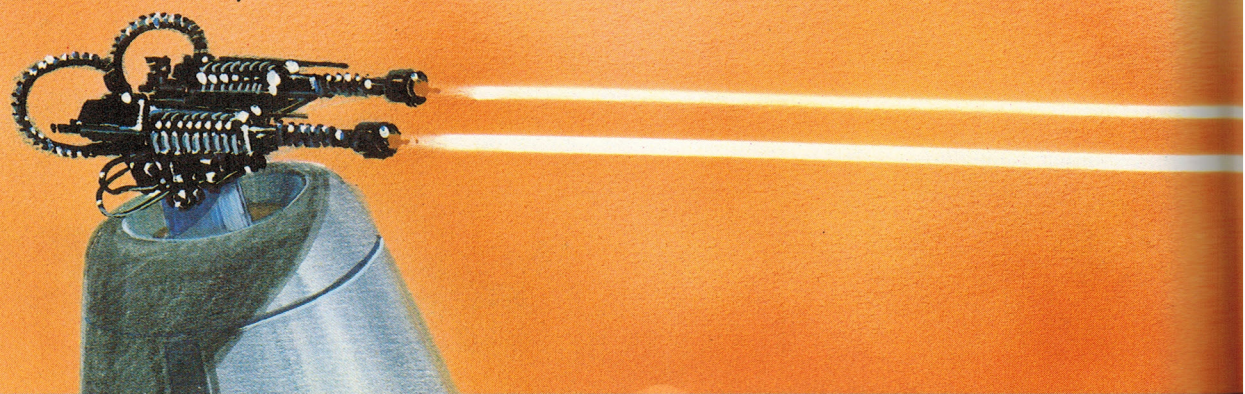
On this shuttle mission you find your ship in a very heavy meteor shower. There is nowhere to land to get out of it, so you'll just have to weather the storm as best you can. You can rotate your ship left and right using keys 1 and 2, and you can move forward by using key 9. Destroy meteors in your path with the zero key, your laser cannon. You may also jump into hyperspace momentarily with the space bar, but each hyperspace jump costs you 100 points! When your score reaches 300, watch out for flying saucers!

VARIABLES USED:

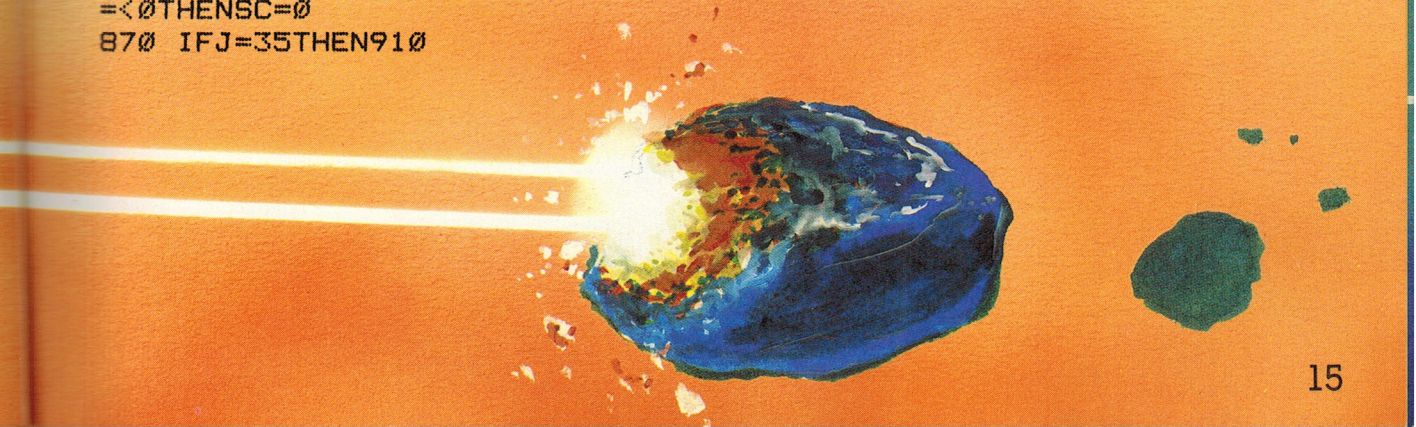
A, AA,
I, Z – general purpose counters
B – data handler
SC – score
F, F2, X – ship position
D(1) –
D(8), X2 – ship direction
M(1) –
M(4), R(1)
– R(4) – asteroid positions
M – random number
T(1) –
T(4), U – UFO positions
J – keyboard
RE$, A$ – keyboard response



```
100 PRINT"{SC}PLEASE WAIT"
110 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
120 FORI=0TO2048:POKE12288+I,PEEK(53248+I):NEXTI
130 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
140 REM MUSIC DATA
150 DATA 3,244,2,3,244,2,3,244,2,5,71,8,7,233,8,0,0,4
160 DATA 7,12,2,6,167,2,5,237,2,10,143,8,7,233,8
170 DATA 0,0,4,7,12,2,6,167,2,5,237,2,10,143,8,7,233,8
180 DATA 7,12,2,6,167,2,7,12,2,5,237,12,0,0,4
190 REM CUSTOM CHAR DATA
200 DATA 16,16,16,16,16,16,16,16,0,0,0,255,0,0,0,0
210 DATA 64,0,1,16,0,64,1,130,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0
220 DATA 68,108,124,40,56,16,16,0,8,12,14,28,252,122,38,1
230 DATA 0,224,120,46,120,224,0,0,1,38,122,252,28,14,12,8
240 DATA 0,16,16,56,40,124,108,68,128,100,95,63,56,112,48,16
250 DATA 0,7,30,116,30,7,0,0,16,48,112,56,63,95,100,128
260 DATA 1,2,4,8,16,32,64,128,128,64,32,16,8,4,2,1
270 DATA 60,60,126,65,255,255,60,60,12,28,30,127,255,255,252
,56
280 DATA 0,8,24,56,48,32,0,0,0,0,56,252,255,255,127
290 REM MOVEMENT DATA
300 DATA40,41,1,-39,-40,-41,-1,39
310 FORI=54272TO54296:POKEI,0:NEXTI
320 POKE54296,15
330 POKE54277,64:POKE54278,128:POKE54276,33
340 POKE51,0:POKE52,48:POKE55,0:POKE56,48
350 POKE53280,0:POKE53281,1
360 PRINT"{SC}":PRINTTAB(13);"{CD}{BL}*METEOR SHOWER*"
370 PRINTTAB(5);"{CD}BY ANDREW STANFORD & PETER J LEAR"
380 PRINTTAB(14);"{CD}{CD}{CD}{CD}<< CONTROLS >>"
390 PRINTTAB(12);"{CD}1 AND 2 TURNS SHIP"
400 PRINTTAB(15);"{CD}9 MOVES SHIP"
410 PRINTTAB(14);"{CD}0 FIRES A BEAM"
420 PRINT"{CD}{CD} TO ENTER HYPERSPACE USE THE SPACE BAR"
430 FORZ=1TO22:READA,B,C:POKE54273,A:POKE54272,B:POKE54276,3
3
440 FORX=1TOC*50:NEXTX:POKE54276,0:NEXTZ
450 POKE54276,33:POKE54272,0:POKE54273,0
460 PRINTTAB(10);"{CD}{CD}PRESS ANY KEY TO START"
470 GETA$:IFA$=""THEN470
480 FORI=232TO382:READB:POKE12288+I,B:NEXTI
520 FORI=1TO8:READDX(I):NEXT
530 POKE53272,(PEEK(53272)AND240)+12
```



```
540 SC=0:X2=34:X=1425
550 POKE0,32
560 PRINT"{SC}":POKE53281,0
570 FORI=1064TO1102:POKEI,33:NEXTI:FORI=1984TO2023:POKEI,33:
NEXTI
580 FORA=1024TO1984STEP40:POKEA,33:NEXT
590 FORA=1TO4
600 FORAA=1TO2
610 IF PEEK(R(A))>33THEN750
620 IFA=1THENR(1)=INT(20*RND(1)+1085):T(1)=INT(3*RND(1)+45):
M(1)=INT(3*RND(1)+39)
630 IFA=1ANDSC>300ANDINT(5*RND(1))=2THENT(1)=44:R(1)=1082:M(
1)=40
640 IFA=2THENR=INT(20*RND(1)):R(2)=1085+(R*40):T(2)=INT(3*RN
D(1)+45):M=INT(6*RND(1))
650 IFA=2ANDM<1THENM(2)=41
660 IFA=2ANDM>1THENM(2)=-41
670 IFA=2ANDSC>300ANDINT(5*RND(1))=2THENT(2)=44:R(2)=X+10:M(
2)=-1
680 IFA=3THENR=INT(20*RND(1)):R(3)=1103+(R*40):T(3)=INT(3*RN
D(1)+45):M=INT(4*RND(1))
690 IFA=3THENM(3)=-39
700 IFA=3ANDM=2THENM(3)=39
710 IFA=3ANDM=1THENM(3)=-1
720 IFA=3ANDSC>300ANDINT(5*RND(1))=2THENT(3)=44:R(3)=X-10:M(
3)=1
730 IFA=4THENR(4)=INT(20*RND(1)+1945):T(4)=INT(3*RND(1)+45):
M(4)=INT(4*RND(1)-41)
740 IFA=4ANDSC>300ANDINT(5*RND(1))=2THENT(4)=44:R(4)=1963:M(
4)=-40
750 POKER(A),32:R(A)=R(A)+M(A)
760 IFT(A)=80THENPOKE54273,100
770 IFR(A)=XTHEN1150
780 POKE54273,0
790 IFPEEK(R(A))<>32THEN820
800 POKER(A),T(A):POKER(A)+54296,A+2
810 NEXTAA
820 J=PEEK(197)
830 IFJ=56THENX2=X2+1:IFX2=42THENX2=34
840 IFJ=59THENX2=X2-1:IFX2=33THENX2=41
850 IFJ=32ANDX+DX(X2-33)>1064ANDX+DX(X2-33)<2023THENPOKEX,32
:X=X+DX(X2-33)
860 IFJ=60THENPOKEX,32:X=1065+INT(RND(1)*900):SC=SC-100:IFSC
=<0THENSC=0
870 IFJ=35THEN910
```



```
880 PRINT"{WH}{HM}{CR}{CR}{CR}{CR}{CR}{CR}{CR}"SC
890 POKEX,X2:POKEX+54296,7
900 NEXT:GOTO590
910 ONX2-33GOTO930,940,950,960,970,980,990,1000
920 GOTO1010
930 X2=34:F=40:F1=29:GOTO1010
940 X2=35:F=41:F1=43:GOTO1010
950 X2=36:F=1:F1=30:GOTO1010
960 X2=37:F=-39:F1=42:GOTO1010
970 X2=38:F=-40:F1=29:GOTO1010
980 X2=39:F=-41:F1=43:GOTO1010
990 X2=40:F=-1:F1=30:GOTO1010
1000 X2=41:F=39:F1=42
1010 F2=X:J=250
1020 F2=F2+F:IFPEEK(F2)<>32THEN1050
1030 POKEF2,F1:POKE54273,J:J=J-2:POKEF2+54296,2
1040 GOTO1020
1050 U=PEEK(F2)
1060 IFPEEK(F2)<34ORPEEK(F2)>47THEN1120
1070 POKEF2,31
1080 POKE54273,150
1090 IFU=44THENSC=SC+50:U=32:GOTO1110
1100 SC=SC+10*A:U=32
1110 PRINT"{WH}{HM}{CR}{CR}{CR}{CR}{CR}{CR}{CR}"SC
1120 FORW=X+FTOF2STEPF:POKEW,32:NEXT
1130 POKE54273,0:POKEF2,U
1140 GOTO890
1150 POKEX,32
1160 POKE54276,129:POKE54273,18
1170 FORI=255TO128
1180 POKE54272,I
1190 FORJ=15TO1STEP-1
1200 POKE54296,J
1210 FORK=1TO100
1220 NEXTK,J,I
1230 POKE54272,0:POKE54296,0:POKE54276,0
1240 FORI=1TO1000:NEXTI
1250 POKE53272,21
1260 PRINT"{SC}{CD}{CD}"
1270 PRINTTAB(10);"PLAY AGAIN (Y/N)"
1280 POKE198,0
1290 GETRE$
1300 IFRE$="Y"THENRUN150
1310 IFRE$="N"THENPRINT"{SC}{BL}":POKE54280,3:POKE54281,6:EN
D
1320 GOTO1290
```

ABOUT DATA STATEMENTS

Meteor Shower and the other games in this book make use of DATA statements. Why DATA statements instead of variables? Because DATA statements can hold information that is not needed all at once or information that is needed only once.

The music in Meteor Shower is only heard at the beginning of the game, so the notes and time durations are stored in lines 150 and 180. The variables are used to hold information that is required very often, for example the asteroid, UFO and ship positions (M(1)-M(4), R(1)-R(4), T(1)-T(4), F, F2, X, X1 and X2).

To use DATA statements in your own programs type "DATA" followed by the information you want stored. Separate each piece by commas.

For example: 100 DATA 8,10,20,–17

Strings may be stored this way too. If a string has punctuation or spaces in it, the string must be enclosed in quotes, otherwise the computer will not know where it ends.

For example:

```
105 DATA FRED,SCORE,BOX,
    "CATS AND MICE", LASER
```

Numerical and string information may be stored in the same DATA statement (up to a total of 80 characters).

For example:

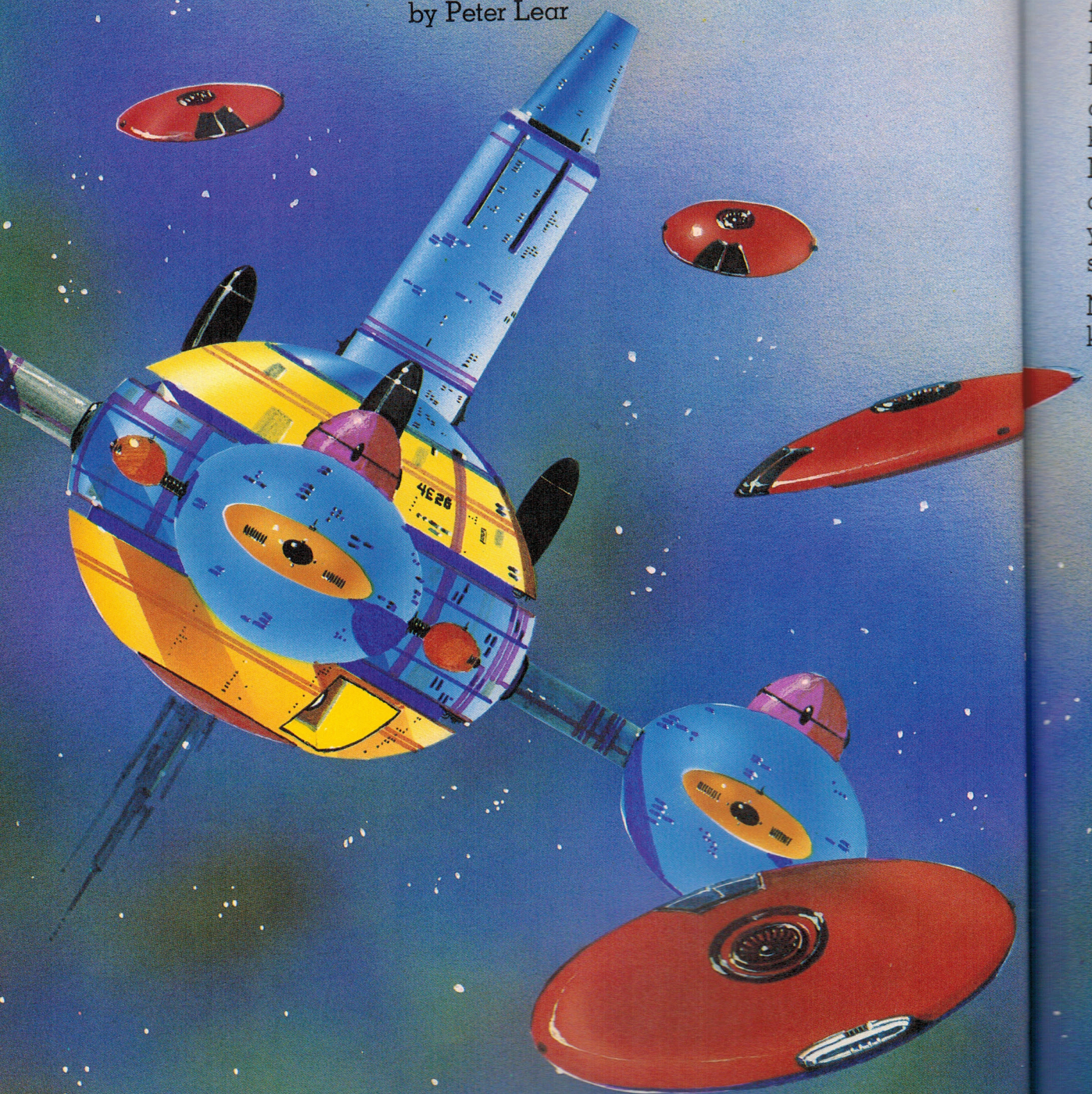
```
110 DATA 255,210,–22,"FUEL
    SUPPLY IS ",ELEPHANT,89
```

If you have more information to store, simply start a new DATA statement.

To retrieve the information stored in a DATA statement, the BASIC command READ is used. To find out more about READ statements, turn to page 29 in this book.

ALIEN ATTACK

by Peter Lear



You await your doom inside the crippled space station. The aliens will be attacking in clusters. There are eight different ways that they can get in. When they do come, the station's force shields will protect you from at least three blasts on each side. Then, if you're lucky you will figure out how to operate the malfunctioning laser cannon and fight back. There are several controls that could do it — your joystick and some keys. Try T, Y, U, G, J, B, N, and M. Maybe by destroying enough attackers you will be able to recharge your shield and save the space station and your life.

Note: Try pressing H, C and other keys after the game.

VARIABLES USED:

JK	– joystick option
I, II, J, K, KK, L	– counters
A$	– general question response
JS(I,J), DD, PB, P,PA,S0,S1, S2,S3,FR	– joystick
DI	– direction of fire
S	– center of shields
M1	– laser origin
P1, P2	– score
C	– screen color
M(1) – M(8)	– enemy ships
D2	– move up
CF, R, A	– flags
D	– difficulty factor
E(1) – E(8)	– explosion data
K1 – K3	– temporary data store
A(1) – A(4), AA(1) – AA(8), B(1) – B(4)	– data handlers
PP	– player character
VOL, V1	– sound control

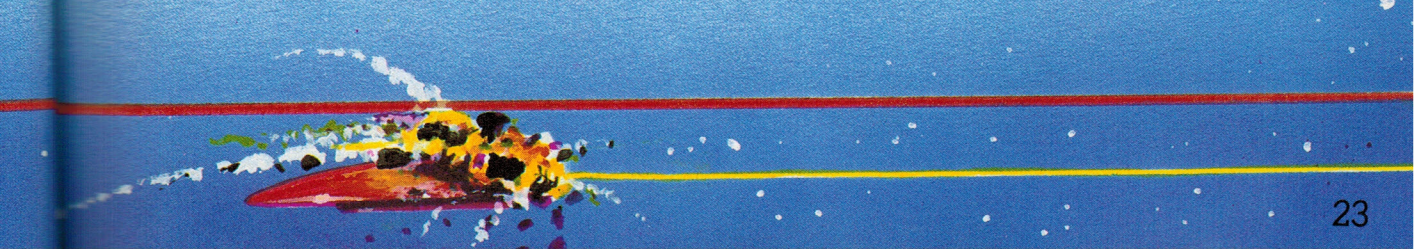
ALIEN ATTACK PROGRAM

```
100 REM ALIEN ATTACK
110 REM BY PETER LEAR
120 REM COPYRIGHT 1982
130 REM BURLINGTON ONT
140 PRINT"{SC}";CHR$(142)
150 POKE53280,0:POKE53281,0
160 READJK,D:IFJK>0THEN230
170 PRINT"{SC}{CD}{CD}":PRINTSPC(15);"ALIEN ATTACK"
180 PRINTSPC(14);"BY PETER LEAR"
190 PRINTSPC(16);"{CD}{CD}{CD}{CD}1-JOYSTICK":PRINTSPC(16);"
2-KEYBOARD"
200 FORI=1TO10:GETA$:NEXTI
210 GETA$:IFVAL(A$)<1ORVAL(A$)>2THEN210
220 JK=VAL(A$)
225 IFJK=2THENGOSUB3360
230 GOTO2200
240 REM JOY CHECK
245 PA=PEEK(56320)
246 J0=-((PAAND2↑0)=0)
247 J1=-((PAAND2↑1)=0)
248 J2=-((PAAND2↑2)=0)
249 J3=-((PAAND2↑3)=0)
250 IFJ0=1ANDJ2=1THENDI=1:RETURN
251 IFJ0=1ANDJ3=1THENDI=3:RETURN
252 IFJ1=1ANDJ2=1THENDI=7:RETURN
253 IFJ1=1ANDJ3=1THENDI=9:RETURN
254 IFJ0=1THENDI=2:RETURN
255 IFJ1=1THENDI=8:RETURN
265 IFJ2=1THENDI=4:RETURN
270 IFJ3=1THENDI=6:RETURN
275 DI=0:RETURN
280 REM KEYBOARD
290 IFPEEK(197)=22THENDI=1:RETURN
300 IFPEEK(197)=25THENDI=2:RETURN
310 IFPEEK(197)=30THENDI=3:RETURN
320 IFPEEK(197)=26THENDI=4:RETURN
330 IFPEEK(197)=36THENDI=9:RETURN
340 IFPEEK(197)=34THENDI=6:RETURN
350 IFPEEK(197)=28THENDI=7:RETURN
360 IFPEEK(197)=39THENDI=8:RETURN
```

```
370 DI=0:RETURN
380 REM -41
390 POKEM1-41,85:POKEM1+C-41,3
400 FORI=S+377TOS+8STEP-41
410 IFPEEK(I)>46THENM(1)=0:P1=P1+50:P2=P2+50:GOTO460
420 POKEI,42:POKEI+C,2:POKEI,32:POKE54296,15
430 NEXTI
440 POKE54296,0
450 RETURN
460 POKES+9,32:POKES+48,32:POKES+49,32
470 POKE54296,0
480 RETURN
490 REM -40
500 POKEM1-40,114:POKEM1+C-40,3
510 FORI=S+260TOS+20STEP-40
520 IFPEEK(I)>46THENM(2)=0:P1=P1+75:P2=P2+75:GOTO570
530 POKEI,42:POKEI+C,2:POKEI,32:POKEVOL,15
540 NEXTI
550 POKEVOL,0
560 RETURN
570 FORI=19TO21:POKES+I,32:NEXTI
580 POKEVOL,0
590 RETURN
600 REM -39
610 POKEM1-39,73:POKEM1+C-39,3
620 FORI=S+383TOS+32STEP-39
630 IFPEEK(I)>46THENM(3)=0:P1=P1+50:P2=P2+50:GOTO680
640 POKEI,42:POKEI+C,2:POKEI,32:POKEVOL,15
650 NEXTI
660 POKEVOL,0
670 RETURN
680 POKES+31,32:POKES+71,32:POKES+72,32
690 POKEVOL,0
700 RETURN
710 REM  -1
720 POKEM1-1,107:POKEM1+C-1,3
730 FORI=S+494TOS+480STEP-1
740 IFPEEK(I)>46THENM(8)=0:P1=P1+75:P2=P2+75:GOTO790
750 POKEI,42:POKEI+C,2:POKEI,32:POKEVOL,15
760 NEXTI
```

```
770 POKEVOL,0
780 RETURN
790 FORI=440TO520STEP40:POKES+I,32:NEXTI
800 POKEVOL,0
810 RETURN
820 POKEM1,90:POKEM1+C,1
830 RETURN
840 REM    1
850 POKEM1+1,115:POKEM1+C+1,3
860 FORI=S+506TOS+519
870 IFPEEK(I)>46THENM(4)=0:P1=P1+75:P2=P2+75:GOTO920
880 POKEI,42:POKEI+C,2:POKEI,32:POKEVOL,15
890 NEXTI
900 POKEVOL,0
910 RETURN
920 FORI=479TO559STEP40:POKES+I,32:NEXTI
930 POKEVOL,0
940 RETURN
950 REM    39
960 POKEM1+39,74:POKEM1+C+39,3
970 FORI=S+617TOS+968STEP39
980 IFPEEK(I)>46THENM(7)=0:P1=P1+50:P2=P2+50:GOTO1030
990 POKEI,42:POKEI+C,2:POKEI,32:POKEVOL,15
1000 NEXTI
1010 POKEVOL,0
1020 RETURN
1030 POKES+928,32:POKES+929,32:POKES+969,32
1040 POKEVOL,0
1050 RETURN
1060 REM    40
1070 POKEM1+40,113:POKEM1+C+40,3
1080 FORI=S+740TOS+980STEP40
1090 IFPEEK(I)>46THENM(6)=0:P1=P1+75:P2=P2+75:GOTO1140
1100 POKEI,42:POKEI+C,2:POKEI,32:POKEVOL,15
1110 NEXTI
1120 POKEVOL,0
1130 RETURN
1140 FORI=1TO3:POKES+979+I,32:NEXTI
1150 POKEVOL,0
1160 RETURN
```

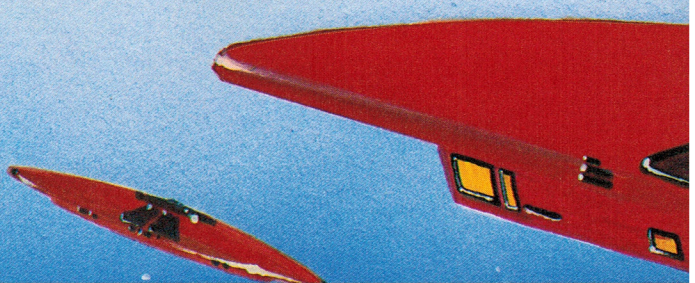
```
117Ø REM    41
118Ø POKEM1+41,75:POKEM1+C+41,3
119Ø FORI=S+623TOS+992STEP41
12ØØ IFPEEK(I)>46THENM(5)=Ø:P1=P1+5Ø:P2=P2+5Ø:GOTO125Ø
121Ø POKEI,42:POKEI+C,2:POKEI,32:POKEVOL,15
122Ø NEXTI
123Ø POKEVOL,Ø
124Ø RETURN
125Ø POKES+951,32:POKES+952,32:POKES+991,32
126Ø POKEVOL,Ø
127Ø RETURN
128Ø REM TOP LEFT
129Ø POKEV1,8:POKEVOL,15
13ØØ POKES+9,1Ø5:POKES+C+9,5:POKES+48,1Ø5:POKES+C+48,5:POKES
+49,75:POKES+C+49,5
131Ø IFM(1)=1THENM(1)=2:POKEV1,9:POKEVOL,Ø:RETURN
132Ø FORI=S+9ØTOS+5ØØSTEP41
133Ø IFI=S+5ØØTHEN27ØØ
134Ø FORII=-1TO1
135Ø IFPEEK(I+II)<>32THENPOKEI+II,32:I=S+5ØØ
136Ø POKEI+II,81:POKEI+II+C,2:POKEI+II,32
137Ø NEXTII,I
138Ø POKEV1,9:POKEVOL,Ø
139Ø RETURN
14ØØ REM TOP MIDDLE
141Ø POKES+19,12Ø:POKES+C+19,5:POKES+2Ø,1Ø2:POKES+C+2Ø,5
1415 POKES+21,12Ø:POKES+C+21,5
142Ø POKEV1,8:POKEVOL,15
143Ø IFM(2)=1THENM(2)=2:POKEV1,9:POKEVOL,Ø:RETURN
144Ø FORI=S+6ØTOS+5ØØSTEP4Ø
145Ø IFI=S+5ØØTHEN27ØØ
146Ø IFPEEK(I)<>32THENPOKEI,32:I=S+5ØØ
147Ø POKEI,81:POKEI+C,2:POKEI,32
148Ø NEXTI
149Ø POKEV1,9:POKEVOL,Ø
15ØØ RETURN
151Ø REM TOP RIGHT
152Ø POKEV1,8:POKEVOL,15
153Ø POKES+31,95:POKES+C+31,5:POKES+71,74:POKES+C+71,5:POKES
+72,95:POKES+C+72,5
```



```
1540 IFM(3)=1THENM(3)=2:POKEV1,9:POKEVOL,0:RETURN
1550 FORI=S+110TOS+500STEP39
1560 IFI=S+500THEN2700
1570 FORII=-1TO1
1580 IFPEEK(I+II)<>32THENPOKEI+II,32:I=S+500
1590 POKEI+II,81:POKEI+II+C,2:POKEI+II,32
1600 NEXTII,I
1610 POKEV1,9:POKEVOL,0
1620 RETURN
1630 REM MIDDLE RIGHT
1640 POKES+479,118:POKES+C+479,5:POKES+519,102:POKES+C+519,5
1645 POKES+559,118:POKES+C+559,5
1650 POKEV1,8:POKEVOL,15
1660 IFM(4)=1THENM(4)=2:POKEV1,9:POKEVOL,0:RETURN
1670 FORI=S+518TOS+500STEP-1
1680 IFI=S+500THEN2700
1690 IFPEEK(I)<>32THENPOKEI,32:I=S+500
1700 POKEI,81:POKEI+C,2:POKEI,32
1710 NEXTI
1720 POKEV1,9:POKEVOL,0
1730 RETURN
1740 REM BOTTOM RIGHT
1750 POKEV1,8:POKEVOL,15
1760 POKES+951,85:POKES+C+951,5:POKES+952,233:POKES+C+952,5
1765 POKES+991,233:POKES+C+991,5
1770 IFM(5)=1THENM(5)=2:POKEV1,9:POKEVOL,0:RETURN
1780 FORI=S+910TOS+500STEP-41
1790 IFI=S+500THEN2700
1800 FORII=-1TO1
1810 IFPEEK(I+II)<>32THENPOKEI+II,32:I=S+500
1820 POKEI+II,81:POKEI+II+C,2:POKEI+II,32
1830 NEXTII,I
1840 POKEV1,9:POKEVOL,0
1850 RETURN
1860 REM BOTTOM MIDDLE
1870 POKES+980,104:POKES+980+C,5:POKES+981,102:POKES+C+981,5
1875 POKES+982,104:POKES+C+982,5
1880 POKEV1,8:POKEVOL,15
1890 IFM(6)=1THENM(6)=2:POKEV1,9:POKEVOL,0:RETURN
1900 FORI=S+940TOS+500STEP-40
```

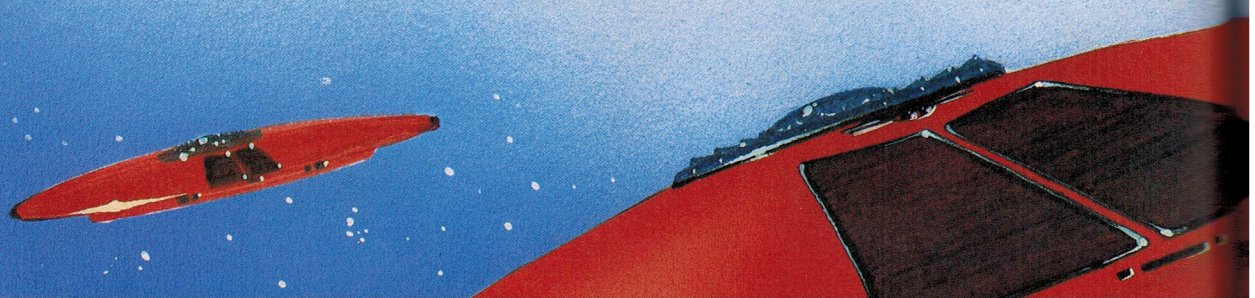
```
1910 IFI=S+500THEN2700
1920 IFPEEK(I)<>32THENPOKEI,32:I=S+500
1930 POKEI,81:POKEI+C,2:POKEI,32
1940 NEXTI
1950 POKEV1,9:POKEVOL,0
1960 RETURN
1970 REM BOTTOM LEFT
1980 POKEV1,8:POKEVOL,15
1990 POKES+928,223:POKES+C+928,5:POKES+929,73:POKES+C+929,5
1995 POKES+969,223:POKES+C+969,5
2000 IFM(7)=1THENM(7)=2:POKEV1,9:POKEVOL,0:RETURN
2010 FORI=S+890TOS+500STEP-39
2020 IFI=S+500THEN2700
2030 FORII=-1TO1
2040 IFPEEK(I+II)<>32THENPOKEI+II,32:I=S+500
2050 POKEI+II,81:POKEI+II+C,2:POKEI+II,32
2060 NEXTII,I
2070 POKEV1,9:POKEVOL,0
2080 RETURN
2090 REM MIDDLE LEFT
2100 POKES+440,92:POKES+C+440,5:POKES+480,102:POKES+C+480,5
2105 POKES+520,92:POKES+C+520,5
2110 POKEV1,8:POKEVOL,15
2120 IFM(8)=1THENM(8)=2:POKEV1,9:POKEVOL,0:RETURN
2130 FORI=S+481TOS+500
2140 IFI=S+500THEN2700
2150 IFPEEK(I)<>32THENPOKEI,32:I=S+500
2160 POKEI,81:POKEI+C,2:POKEI,32
2170 NEXTI
2180 POKEV1,9:POKEVOL,0
2190 RETURN
2200 REM SET UP
2210 S=256*PEEK(648):C=55296-S
2220 DIMJS(2,2):REM JOY VARS
2230 FORI=0TO2:FORJ=0TO2:READJS(J,I):NEXTJ,I
2240 FORI=1TO9:READHS(I):NEXTI
2250 FORI=1TO9:READD(I):NEXTI
2260 M1=S+500:R=7:PP=113:D2=40:CF=1:A=2
2265 VOL=54296:V1=54273
2270 FORI=54272TO54296:POKEI,0:NEXTI
```

```
2272 POKEV1,1:POKE54280,5
2275 POKE54285,64:POKE54277,64:POKE54284,64:POKE54278,64
2276 POKE54283,17:POKE54276,129
2280 IFD>0THEN2330
2290 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}":PRINTSPC(13
);"SHOTS FACTOR (1TO9)"
2300 GETA$
2310 IFVAL(A$)<1THEN2300
2320 D=VAL(A$)
2330 PRINT"{SC}"
2360 FORI=300TO380STEP40:POKES+I,100:POKES+C+I,1:NEXTI
2370 FORI=417TO419:POKES+I,78:POKES+C+I,1:NEXTI
2372 POKEVOL,0:POKEV1,9:POKE54280,5:POKE54272,0:POKE54279,0
2380 FORI=421TO423:POKES+I,77:POKES+C+I,1:NEXTI
2390 FORI=456TO458:POKES+I,78:POKES+C+I,1:NEXTI
2400 FORI=462TO464:POKES+I,77:POKES+C+I,1:NEXTI
2410 FORI=495TO497:POKES+I,103:POKES+C+I,1:NEXTI
2420 FORI=503TO505:POKES+I,101:POKES+C+I,1:NEXTI
2430 FORI=536TO538:POKES+I,77:POKES+C+I,1:NEXTI
2440 FORI=542TO544:POKES+I,78:POKES+C+I,1:NEXTI
2450 FORI=577TO579:POKES+I,77:POKES+C+I,1:NEXTI
2460 FORI=581TO583:POKES+I,78:POKES+C+I,1:NEXTI
2470 FORI=620TO700STEP40:POKES+I,99:POKES+C+I,1:NEXTI
2480 POKES+379,78:POKES+C+379,1:POKES+381,77:POKES+C+381,1
2490 POKES+619,77:POKES+C+619,1:POKES+621,78:POKES+C+621,1
2500 POKES+339,78:POKES+C+339,1:POKES+341,77:POKES+C+341,1
2510 POKES+378,78:POKES+C+378,1:POKES+382,77:POKES+C+382,1
2520 POKES+618,77:POKES+C+618,1:POKES+622,78:POKES+C+622,1
2530 POKES+659,77:POKES+C+659,1:POKES+661,78:POKES+C+661,1
2540 IFP2>0THENRETURN
2550 PRINT"{HM}";SPC(35);"{CU}";HS(D)
2560 TI$="000000"
2570 REM MAIN PLAY
2580 PRINT"{HM}{CR}{CR}{CR}"P2
2590 ONJKGOSUB245,290
2600 ONDIGOSUB390,500,610,720,820,850,960,1070,1180
2610 POKEM1+D(DI),32
2620 IFP1=>1200THENP1=P1-1200:GOSUB2360
2630 A=A*2:IFA<D*2THEN2580
2640 FORJ=1TOINT(P2/1000)STEP2
```



```
2650 M=INT(RND(TI)*(12+5*INT(P2/1000)))
2655 IFM>2+5*INT(P2/1000)THENM((12+5*INT(P2/1000))-M)=1
2660 FORM2=1TO8
2670 IFM(M2)>0THENONM2GOSUB1290,1410,1520,1640,1750,1870,198
0,2100
2680 NEXTM2,J
2690 A=2:GOTO2580
2700 REM EXPLOSION
2710 A$=MID$(TI$,4,1)+":"+RIGHT$(TI$,2)
2720 FORII=1TO8:E(II)=S+540:READA(II):NEXTII
2730 FORII=1TO8:E(II)=S+540:READB(II):NEXTII
2740 POKE54280,0:POKEV1,0:POKEVOL,15
2750 FORJ=1TO5
2760 FORII=1TO8:E(II)=A(II)+E(II):NEXTII
2770 FORII=1TO8:POKEE(II),32:NEXTII
2780 FORII=1TO8:POKEE(II),B(II):NEXTII
2790 POKEV1,5
2800 NEXTJ
2810 FORJ=1TO5
2820 FORII=1TO8:POKEE(II),32:NEXTII
2830 FORII=1TO8:E(II)=E(II)-A(II):NEXTII
2840 NEXTJ
2850 FORK=7TO18
2860 FORKK=15TO21
2870 POKES+K*40+KK,32:POKES+29*40-K*40-KK,32
2880 NEXTKK,K
2910 E(1)=S+499:E(2)=S+501:E(3)=S+541:E(4)=S+539
2920 FORII=1TO4:READA(II),B(II):NEXTII
2930 FORII=1TO8:READAA(II):POKE53281,AA(II):FORG=1TO25:NEXTG
,II
2940 POKEV1,4:POKE53281,0
2950 FORL=15TO0STEP-3
2960 FORII=1TO4:E(II)=E(II)-A(II):POKEE(II),B(II):NEXTII
2970 POKEVOL,L
2990 FORII=1TO4:POKEE(II),32:NEXTII
3000 NEXTL
3010 POKEVOL,0
3020 IFP2>HS(D)THENHS(D)=P2
3030 I=0:II=0:J=0:KK=0:M2=0
3040 PRINT"{HM}{CR}{CR}{CR}"P2;SPC(13);"{CD}{CD}";A$:P2=0:P1
=0
```

```
3050 PRINT"{HM}";SPC(35);"{CU}"HS(D)
3060 POKE198,0
3070 GETA$
3090 IFA$="H"THENGOSUB3240
3100 IFA$="C"THEND=0:GOTO3140
3110 IFA$<>""THEN3140
3120 IF((PEEK(56320)AND2↑4)=0)=-1THEN3140
3130 GOTO3070
3140 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}"
3150 K=2:K1=89:K3=56
3160 PRINT"3410 DATA";JK;"{CL},";D
3170 PRINT"3430 DATA";
3180 FORK=1TO9:PRINTHS(K)"{CL},";:NEXTK:PRINTCHR$(20)
3190 PRINT"RUN"
3200 PRINT"{HM}"
3210 POKE198,10
3220 FORI=631TO640:POKEI,13:NEXTI
3230 STOP
3240 PRINT"{SC}{CD}"
3250 PRINTSPC(10);"LEVEL   ";" HI SCORE":PRINT
3260 FORK=1TO9
3270 PRINTSPC(10);K,HS(K)
3280 PRINT
3290 NEXTK
3300 FORJ=1TO10:GETA$:NEXTJ
3310 GETA$
3330 IFA$<>""THENRETURN
3340 REM
3350 GOTO3310
3360 PRINT"{SC}{CD}{CD}{CD}":PRINTSPC(13);"USE THESE KEYS"
3370 PRINTSPC(18);"{CD}T Y U"
3380 PRINTSPC(18);"{CD}G   J"
3390 PRINTSPC(18);"{CD}B N M"
3400 GOSUB3300
3407 RETURN
3410 DATA 0, 0
3420 DATA1,2,3,4,5,6,7,8,9
3430 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
3440 DATA-41,-40,-39,-1,0,1,39,40,41
3450 DATA-41,-40,-39,1,41,40,39,-1
3460 DATA77,66,78,64,77,66,78,64
3470 DATA39,77,41,78,-41,78,-49,77
3480 DATA1,7,2,1,4,3,7,1
```



ABOUT READ STATEMENTS

To retrieve the information stored in a DATA statement, the BASIC command READ is used. It is followed by one or more variables separated by commas. READ takes the first piece of information from the first DATA statement and puts it into the variable that follows the READ statement. If there is a second variable, the next piece of information is taken from the same DATA statement and put into the second variable. This continues until all the variables following the READ statement are filled. If everything from the first DATA statement is read, the computer automatically jumps to the next one.

There must be as many pieces of information in your DATA statements as are to be READ. The variable type (string or number) must match the type of information being READ. Here is an example using the DATA statements from page 17:

120 READ N This line reads the number 8 from the DATA statement in line 100 and puts it in the variable N.

130 READ A,B,C,D$,E$,F$,G$
This line reads all the information in line 100 and some from 105. Notice how the string variable reads string information and the number variables read numbers.

140 READ H$,I,J,K,L$,M$,O
This line reads all the remaining information in lines 105 and 110.
If information in a DATA statement does not match the type being READ, you will get a "TYPE MISMATCH" error.

When two commas follow one another without any information between them, a READ statement will take this as either an empty string or the number zero. Here is an example:

```
150 DATA 10,20,,15,GREG,,6,7
160 READ G,H,I,J,K$,L$,M,N
```

This program gives the variables the following information:

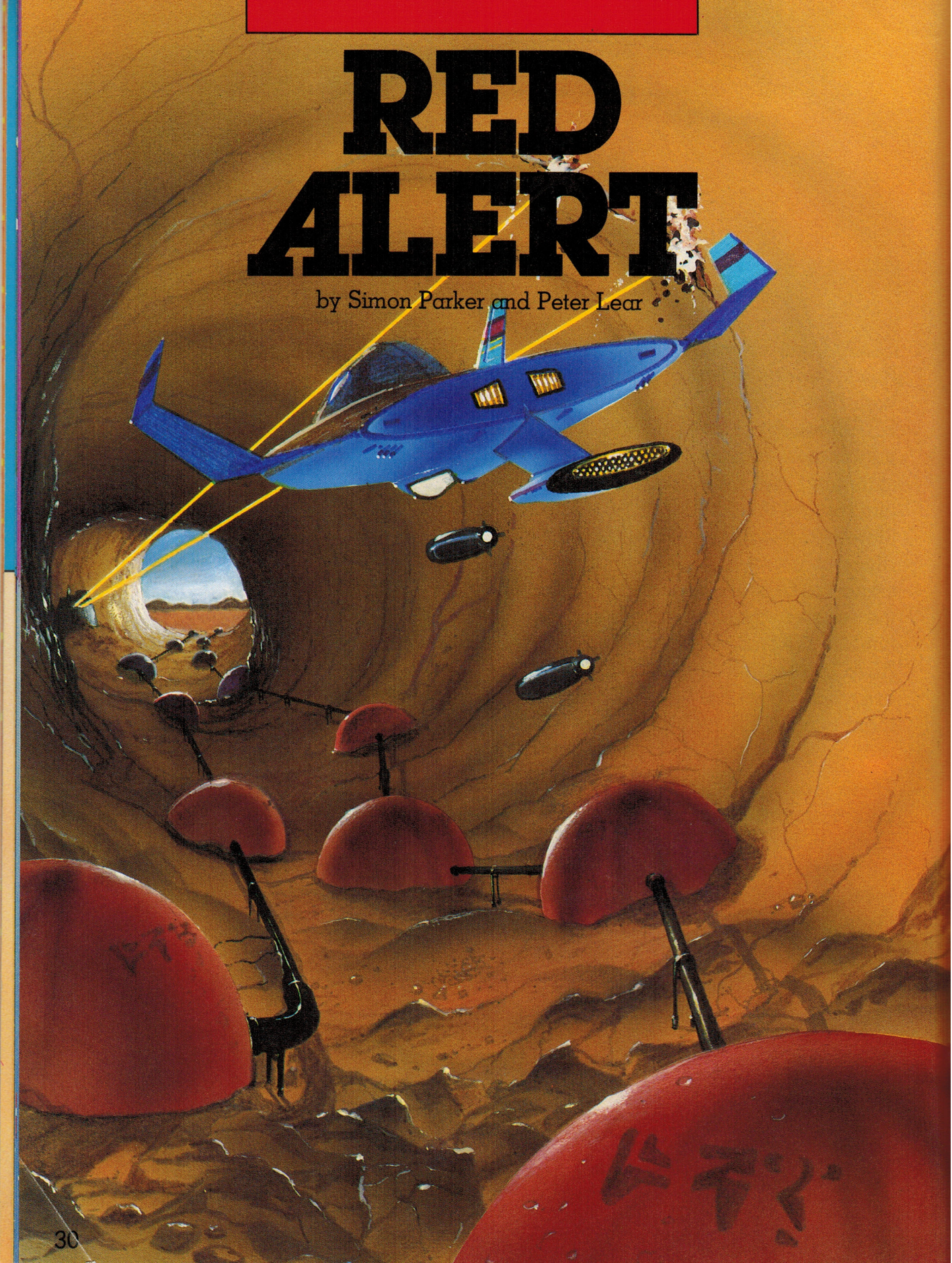
G=10
H=20
I=0
J=15
K$="GREG"
L$="" This is an empty string.
M=6
N=7

When a DATA statement has been totally READ, another READ command will result in an "OUT OF DATA ERROR". To avoid this, always match up the number of pieces of information with the number of variables that will READ them.

The RESTORE command will allow you to use the DATA statements again.

RED ALERT

by Simon Parker and Peter Lear



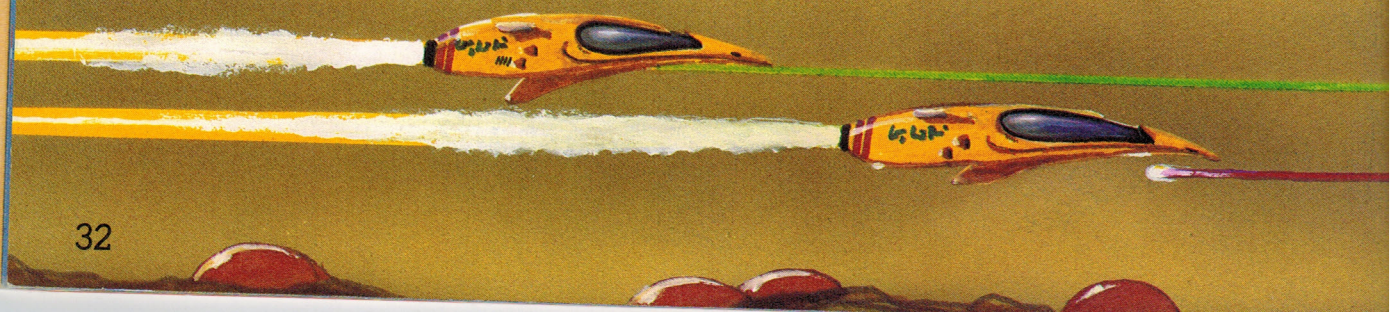
You have penetrated the enemy planet's refueling station. You must pilot your spacecraft through the cavern that houses the fuel dump. The terrain is rough and changes radically. Be very careful; the survival of your planet depends on you. Use the @ key to move your ship up, and the ; key to move down; P moves you left and * moves you right. Bomb the fuel dumps with the shift key. You gain 200 points for every exploded fuel tank and lose 50 points for dropping a bomb and missing one. Oh, and do avoid the counter-attacking aliens. Shoot them with your laser gun (key A) and get 50 points for every "kill"; however, if you can bomb an alien ship you will gain 250 points! Good hunting!

VARIABLES USED:

P	– ship position
G	– top of cavern
J	– bottom of cavern
CC, DD	– cavern height multiplying factors
B, C	– random numbers
K, I, L, M, Z	– counters
L2, P2	– key pressed
S, Y, A, A$	– temporary data handlers
N, N1	– bomb position
SC	– score

RED ALERT PROGRAM

```
100 FORI=54272TO54296:POKEI,0:NEXTI
110 POKE53280,7:POKE53281,5
120 PRINT"{SC}":PRINTSPC(15);"{RD}RED ALERT"
130 PRINTSPC(12);"{CD}{BK}BY SIMON PARKER"
140 PRINTSPC(12);"AND PETER J LEAR"
150 PRINTSPC(5);"{CD}{CY}TAKE YOUR SHIP THROUGH THE CAVE"
160 PRINTSPC(6);"{CD}HIT TANKS AND JETS FOR POINTS"
170 PRINTSPC(14);"{CD}{WH}USE THE KEYS:":PRINTSPC(14);"{CD}@
 -UP"
180 PRINTSPC(14);";  -DOWN":PRINTSPC(14);"P -LEFT"
190 PRINTSPC(14);"* -RIGHT":PRINTSPC(14);"A -LASER"
200 PRINTSPC(14);"SHIFT -BOMB"
210 PRINTSPC(14);"{CD}{YL}PLEASE WAIT"
220 P=1429:J=5:G=5:CC=.5:DD=.9
230 GOSUB650:GOSUB1160
240 TI$="000000"
250 PRINT"{SC}{WH}"
260 POKE54277,128:POKE54278,128:POKE54284,64:POKE54285,64
270 POKE53280,0:POKE53281,0:POKE54296,15:POKE54276,129:POKE5
4273,20
280 POKE54279,100:POKE54280,18
290 B=RND(1)
300 IFTI$="000200"THENPOKE53281,7:CC=.65:DD=.8
310 IFTI$="000330"THENPOKE53281,3:CC=.75:DD=.7
320 C=RND(1)
330 IFB<CCTHENJ=J+1:GOTO350
340 J=J-1
350 IFC<CCTHENG=G+1:GOTO370
360 G=G-1
370 IFJ>0ANDG>10THENG=G-1:J=J-1
380 IFJ+G>17THENJ=J-1:G=G-1
390 IFJ<1THENJ=1
400 IFG<1THENG=1
410 FORK=1TOJ
420 POKE2023-40*K,20
430 NEXTK:IFB<.2ORB>.8THENPOKE2023-40*K,22:POKE56295-40*K,2
440 FORK=1TOG
```

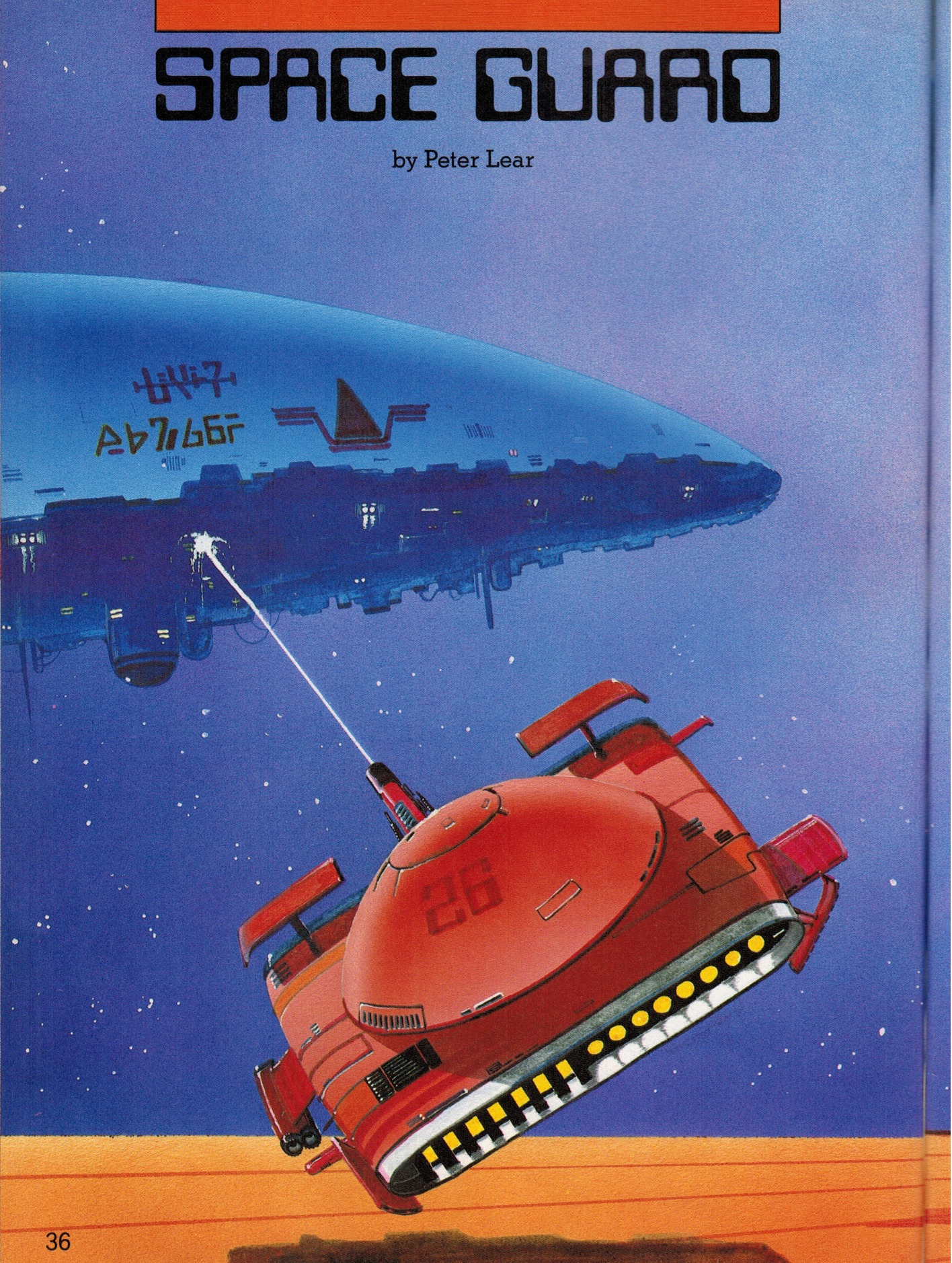


```
450 POKE1063+40*K,20
460 NEXTK
470 IFC>DDTHENPOKE1063+40*K+80,24:POKE55335+40*K+80,2
480 POKEP,32
490 SYS(829)
500 POKEP,21:POKEP-1,32
510 PRINT"{HM}";SC
520 POKEP,32
530 L2=PEEK(197):P2=PEEK(653)
540 IFL=30ANDP1=119ORP2=1ANDL2=49THENGOSUB1000:GOTO610
550 IFL=30ORP2=1THENSC=SC-50:GOSUB930:GOTO610
560 IFL2=10THENGOSUB1060
570 IFL=54ORL2=50THENP=P+40
580 IFL=46ORL2=41THENP=P-1
590 IFL=58ORL2=46THENP=P-40
600 IFP1=119ORL2=49THENP=P+1
610 IFPEEK(P+1)<>32THENGOTO770
620 POKEP,21:POKEP-1,32
630 POKE54272,120+INT((P-1024)/8)
640 GOTO290
650 POKE52,48:POKE56,48
660 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
670 FORI=12288TO12799:POKEI,PEEK(I+40960):NEXTI
680 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
690 FORI=12448TO12487:READA:POKEI,A:NEXTI
700 POKE53272,(PEEK(53272)AND240)+12
710 RETURN
720 DATA170,85,170,85,170,85,170,85
730 DATA0,128,192,224,254,255,255,0
740 DATA60,60,126,126,255,60,66,129
750 DATA160,80,168,75,168,80,160,0
760 DATA0,1,3,7,127,255,255,0
770 FORI=1TO10
780 POKE53281,I*10
790 POKE54296,10-I
800 POKE54276,129:POKE54273,10
810 FORK=1TO200:NEXTK
820 NEXTI
830 FORL=1TO1000:NEXTL
840 POKE54276,0
```

```
850 POKE53272,21:PRINT"{SC}"
860 PRINTSPC(15);"{CD}{CD}{RV}GAME OVER{RO}"
870 PRINTSPC(12);"{CD}{CD}YOU SCORED"SC
880 PRINTSPC(10);"{CD}{CD}ANOTHER GAME (Y/N)"
890 GETA$:IFA$=""THEN890
900 IFA$="Y"THENJ=8:G=8:P=1429:SC=0:TI$="000000":POKE53272,2
9:GOTO250
910 IFA$="N"THENPRINT"{SC}":POKE53272,21:END
920 GOTO890
930 POKE54283,33
940 FORM=1TO10
950 POKEN,32
960 N=P+M*40:POKE54279,150+M*10
970 IFPEEK(N)=22THENSC=SC+250:POKEN,42:POKE54283,0:RETURN
980 IFPEEK(N)=20THENPOKEN,42:POKE54283,0:RETURN
990 POKEN,90:NEXTM:POKEN,32:POKE54283,0:RETURN
1000 SC=SC-100:FORB=1TO10:POKE54283,17:POKEM1,32:M1=P+B:POKE
54283,0
1010 IFPEEK(M1)=20THENPOKEM1,20:RETURN
1020 IFPEEK(M1)=22THENSC=SC+200:POKEM1,32:RETURN
1030 IFPEEK(M1)=24THENSC=SC+150:POKEM1,32:RETURN
1040 POKEM1,42:POKEM1+54296,1
1050 NEXTB:POKEM1,32:RETURN
1060 POKE54283,17
1070 FORZ=P+2TOP+7
1080 Y=PEEK(Z)
1090 POKEZ,46
1100 IFY=24THENSC=SC+50:Y=32
1110 IFY=22THENSC=SC+200:Y=32
1120 POKEZ,Y
1130 NEXTZ
1140 POKE54283,0
1150 RETURN
1160 FORK=829TO874:READS:POKEK,S:NEXTK:RETURN
1170 DATA169,19,32,210,255,169,29,32,210,255,169,13,32,210,2
55,169,0,141,60,3
1180 DATA169,29,32,210,255,169,20,32,210,255,169,13,32,210,2
55,238,60,3,173,60,3
1190 DATA201,23,208,231,96
1200 RETURN
```

SPACE GUARD

by Peter Lear



(Joystick required)

You are Captain Buck Dodger in outer space and you have a mission. You are to protect your star system's satellites from invading aliens. The aliens are trying to alter your satellites to their own means. All satellites in any sector that the aliens have been in must be recovered by you. Be cautious of the aliens. They travel in groups of three. One ship is slow, another is fast and the third is similar in speed to your own ship. If the aliens corner you, press the hyperspace button on your joystick.

Special note: This game consists of two programs. The first sets up some machine language subroutines to give the game its speed. These routines are stored in DATA statements. Enter the first program and SAVE it. Each DATA statement contains eight numbers. As there are a lot of numbers to type in, we have a test to check that you entered the right numbers. After entering the first program, type in this line and press return:

```
B=0: FOR I=0 TO 79: READ
A:B=B+A: NEXT I: PRINT B
```

The computer will pause and then give you a number. Make a note of the number. Use the cursor keys to erase the number and go back up to the line and press return. The computer will give you another number (also note this number). Do this until you have nine numbers. Compare the numbers to this chart:

TRIAL	NUMBER	LINES
1	8611	1-10
2	11019	11-20
3	11398	21-30
4	11529	31-40
5	11161	41-50
6	11016	51-60
7	11132	61-70
8	10113	71-80
9	6010	81-90

If one or more of the numbers you got does not match the chart, check the line numbers in the chart next to that number. To do the test again type in:

```
RESTORE
```

If you are using cassette instead of disk, change line 109 to read:
109 PRINT "[SC] [CD] [CD] [CD] [CD] LOAD"; CHR$ (34); "SPACE GUARD"; CHR$ (34)

Once you have a correct copy SAVE it with the title "SETUP SPACE".

Next enter the second program. When you have the second program entered, SAVE a copy with the name "SPACE GUARD". If you are using cassette SAVE the second program right after the first program.

When you are ready to play the game, LOAD the first program and RUN it. The second program will start automatically.

If the game stops abruptly without any error messages from the computer, go back and check the first program's DATA statements. Even though the numbers from the test may have matched, two (or more) numbers may be switched. Also check the numbers not in DATA statements match those in the book.

Make sure you SAVE a final copy of each program!

This sounds like a lot of work, but it is going to be worth it.

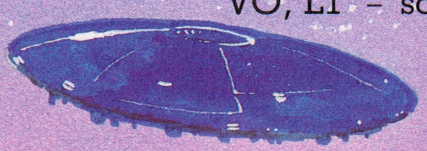
VARIABLES USED:

SETUP PROGRAM

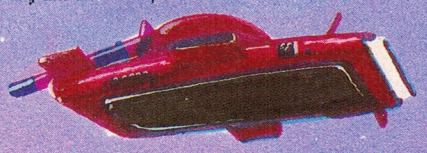
CC – start of machine language pointer
I, J – counters
A – data holder

MAIN PROGRAM

RE$, RE – keyboard response
MIN – minimum number from keyboard
MAX – maximum number from keyboard
T – delay time
P – satellite position
SC – character screen
CO – color screen
RAN – random number
I, J, K – counters
P1 – next ship
PP – score
BO – board number
L – level
HY – hyperspace jumps
MN – ships remaining
S – sector
HS – high score
HS$ – high score initials
P – ship position
EXC (1-9) – explosion colors
N, D1, D2, A, B – temporary number holders
S – flag
SH, SL, SV – enemy ship size
SH – enemy sprite selector
H1, V1, VO, L1 – sound control

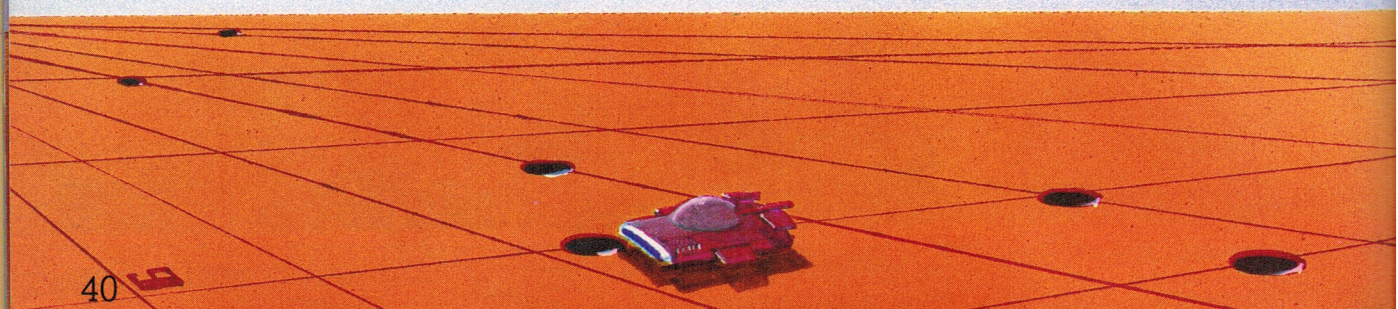


```
1 DATA173,0,220,133,251,234,74,176
2 DATA16,74,74,176,3,76,64,64
3 DATA74,176,3,76,128,64,76,192
4 DATA64,165,251,74,74,176,15,74
5 DATA176,3,76,211,64,74,176,3
6 DATA76,19,65,76,80,65,165,251
7 DATA74,74,74,176,3,76,102,65
8 DATA74,176,3,76,149,65,96,234
9 DATA162,2,206,1,208,169,31,205
10 DATA1,208,208,3,238,1,208,173
11 DATA16,208,74,144,22,206,0,208
12 DATA169,255,205,0,208,208,8,169
13 DATA254,45,16,208,141,16,208,202
14 DATA208,216,96,206,0,208,169,7
15 DATA205,0,208,208,3,238,0,208
16 DATA202,208,199,96,234,234,234,
   234
17 DATA162,2,206,1,208,169,31,205
18 DATA1,208,208,3,238,1,208,173
19 DATA16,208,74,144,17,238,0,208
20 DATA169,80,205,0,208,208,3,206
21 DATA0,208,202,208,221,96,238,0
22 DATA208,169,255,205,0,208,208,8
23 DATA169,1,13,16,208,141,16,208
24 DATA202,208,199,96,234,234,234,
   234
25 DATA162,2,206,1,208,169,31,205
26 DATA1,208,208,3,238,1,208,202
27 DATA208,240,96,162,2,238,1,208
28 DATA169,246,205,1,208,208,3,206
29 DATA1,208,173,16,208,74,144,22
30 DATA206,0,208,169,255,205,0,208
31 DATA208,8,169,254,45,16,208,141
32 DATA16,208,202,208,216,96,206,0
33 DATA208,169,7,205,0,208,208,3
34 DATA238,0,208,202,208,199,96,234
35 DATA234,234,234,162,2,238,1,208
36 DATA169,246,205,1,208,208,3,206
37 DATA1,208,173,16,208,74,144,17
38 DATA238,0,208,169,80,205,0,208
39 DATA208,3,206,0,208,202,208,221
40 DATA96,238,0,208,169,255,205,0
41 DATA208,208,8,169,1,13,16,208
42 DATA141,16,208,202,208,199,96,
   234
43 DATA162,2,238,1,208,169,246,205
44 DATA1,208,208,3,206,1,208,202
45 DATA208,240,96,234,234,234,162,2
46 DATA173,16,208,74,144,22,206,0
47 DATA208,169,255,205,0,208,208,8
48 DATA169,254,45,16,208,141,16,208
49 DATA202,208,229,96,206,0,208,169
50 DATA7,205,0,208,208,3,238,0
51 DATA208,202,208,212,96,162,2,173
52 DATA16,208,74,144,17,238,0,208
53 DATA169,80,205,0,208,208,3,206
54 DATA0,208,202,208,234,96,238,0
55 DATA208,169,255,205,0,208,208,8
56 DATA169,1,13,16,208,141,16,208
57 DATA202,208,212,96,0,0,0,0
58 DATA166,252,164,253,222,1,208,169
59 DATA31,221,1,208,208,3,254,1
60 DATA208,136,208,240,96,166,252,164
61 DATA253,254,1,208,169,246,221,1
62 DATA208,208,3,222,1,208,136,208
63 DATA240,96,164,253,166,252,189,59
64 DATA3,45,16,208,221,59,3,208
65 DATA28,222,0,208,169,0,221,0
66 DATA208,208,14,189,59,3,77,16
67 DATA208,141,16,208,169,255,157,0
68 DATA208,136,96,208,215,222,0,208
69 DATA169,7,221,0,208,208,3,254
70 DATA0,208,136,208,199,96,164,253
71 DATA166,252,189,59,3,45,16,208
72 DATA221,59,3,208,21,234,234,254
73 DATA0,208,169,80,221,0,208,208
74 DATA3,222,0,208,136,208,225,96
75 DATA234,234,254,0,208,169,255,221
76 DATA0,208,208,14,189,59,3,13
77 DATA16,208,141,16,208,169,1,157
78 DATA0,208,136,208,195,96,0,0
79 DATA166,252,173,0,208,221,0,208
80 DATA176,6,32,42,66,24,144,3
81 DATA32,102,66,173,1,208,221,1
82 DATA208,176,6,32,0,66,24,144
83 DATA3,32,21,66,96,166,252,173
84 DATA16,208,74,144,21,189,59,3
85 DATA45,16,208,221,59,3,208,4
86 DATA32,0,67,96,32,102,66,24
87 DATA144,21,189,0,3,45,16,208
88 DATA221,59,3,208,6,32,42,66
89 DATA24,144,4,32,0,67,96,32
90 DATA19,67,96,0,0,0,0,0
91 PRINT"{SC}PLEASE WAIT"
92 POKE56,52:POKE55,0:POKE52,52:POKE51,0
93 CC=16384
94 FORI=0TO56
95 FORJ=0TO7
96 READA
97 POKECC+I*8+J,A
98 NEXTJ,I
99 FORI=64TO84
100 FORJ=0TO7
101 READA
102 POKECC+I*8+J,A
103 NEXTJ,I
104 FORI=96TO107
105 FORJ=0TO7
106 READA
107 POKECC+I*8+J,A
108 NEXTJ,I
109 PRINT"{SC}{CD}{CD}{CD}{CD}LOAD";
CHR$(34);"SPACE GUARD";CHR$(34);",8"
110 PRINT"{CD}{CD}{CD}{CD}{CD}{CD}
{CD}RUN{HM}"
111 FORI=631TO640:POKEI,13:NEXTI
112 POKE198,10
```

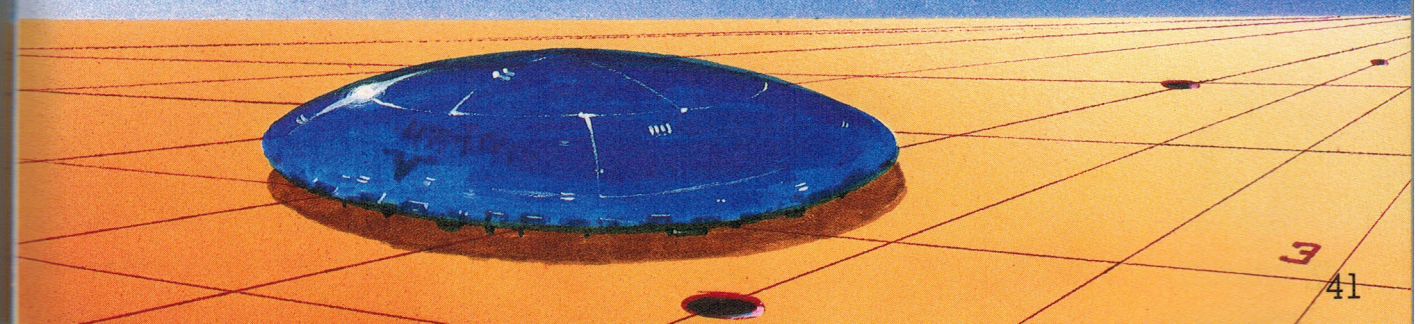


SPACE GUARD PROGRAM

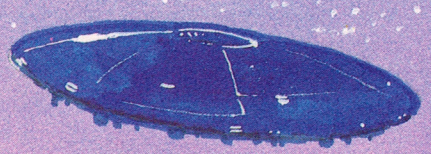
```
100 PRINT"{SC}"
110 PRINTSPC(15);"SPACE GUARD"
120 PRINTSPC(15);"{CD}{YL}PLEASE WAIT"
130 POKE56,52:POKE52,52:POKE55,0:POKE51,0
140 POKE53280,0:POKE53281,6:POKE646,1
150 POKE53269,0
160 GOTO2330
170 REM GET KEYBOARD RESPONSE
180 POKE198,0
190 GETRE$:IFRE$=""THEN190
200 RETURN
210 GOSUB170
220 RE=VAL(RE$)
230 IFRE<MINORRE>MAXTHEN210
240 RETURN
250 REM DELAY
260 FORI=1TOT*1000:NEXTI
270 RETURN
280 POKEH1,10:POKEV1,17:POKEVO,10
290 FORM=1TO50:NEXTM
300 RETURN
310 POKEV1,17:POKEVO,10
320 FORK=1TO11
330 POKEL1,FNRAN(256)-1:POKEH1,FNRAN(256)-1
340 FORM=1TO30:NEXTM
350 NEXTK
360 POKEVO,0:POKEH1,0:POKEV1,0:POKEL1,0
370 RETURN
380 REM SETUP BOARD
390 FORI=1TO10
400 P=FNRAN(1000)
410 IFPEEK(P+1024)<>32THEN400
420 POKEP+SC,42:POKEP+CO,14
430 NEXTI
440 RETURN
450 FORI=1TO10
460 P=FNRAN(1000)
470 IFPEEK(P+1024)<>32THEN460
480 POKEP+SC,90:POKEP+CO,8
490 NEXTI
500 RETURN
510 FORI=1TO10
520 P=FNRAN(1000)
530 IFPEEK(P+1024)<>32THEN520
```



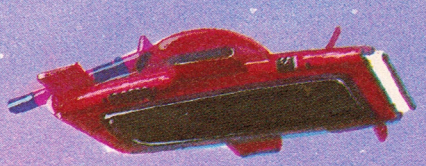
```
540 POKEP+SC,35:POKEP+CO,10
550 NEXTI
560 RETURN
570 FORI=1TO10
580 P=FNRAN(1000)
590 IFPEEK(P+1024)<>32THEN580
600 POKEP+SC,81:POKEP+CO,13
610 NEXTI
620 RETURN
630 REM SECTOR COMPLETE
640 POKE53269,0:POKE53281,4
650 POKEV0,15:POKEV1,33
660 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}"
670 PRINTSPC(12);"{YL}SECTOR COMPLETE"
680 PRINTSPC(13);"{CD}{CD}{RD}YOUR SCORE IS{BK}"
690 FORI=1TO10
700 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{C
D}";SPC(18);"              "
710 PRINT"{CU}";SPC(18);PP+I*BO*25
720 FORJ=1TO100:NEXTJ
730 POKEH1,10+I
740 NEXTI
750 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}"
760 PRINTSPC(16);"{RD}BONUS OF{BK}"
770 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}";
SPC(18);L*100
780 FORJ=1TO500:NEXTJ
790 POKEV1,0:POKEH1,0:POKEL1,0
800 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}"
810 PRINTSPC(14);"{YL}TOTAL SCORE{BK}"
820 PP=PP+10*25*BO+L*100
830 IFPP-P1=>10000THENMN=MN+1:P1=P1+10000
840 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}";
SPC(18);PP
850 PRINTSPC(7);"{CD}{CD}{WH}YOU HAVE";HY;" HYPERSPACE JUMPS
"
860 PRINTSPC(10);"{CD}AND";MN;" SHIPS REMAINING"
870 PRINTSPC(17);"{CD}{YL}LEVEL{WH}";L
880 POKEV1,17:POKEH1,15:POKEV0,15
890 FORJ=1TO10
900 FORK=1TO50:POKEH1,K:NEXTK
910 FORK=50TO1STEP-1:POKEH1,K:NEXTK,J
920 POKE53281,6
930 POKEV0,0:POKEV1,0:POKEH1,0:POKEL1,0
940 RETURN
```



```
950 REM BOARD NOT COMPLETE
960 POKE53281,7:POKE53269,0
970 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}"
980 PRINTSPC(17);"{PU}SCORE{BK}"
990 PP=PP+S*25*BO
1000 IFPP-P1=>10000THENMN=MN+1:P1=P1+10000
1010 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}"
;SPC(17);PP
1020 PRINTSPC(7);"{CD}{CD}{WH}YOU HAVE";HY;" HYPERSPACE JUMP
S"
1030 PRINTSPC(10);"{CD}AND";MN;" SHIPS REMAINING"
1040 PRINTSPC(16);"{CD}{PU}LEVEL{WH}";L
1050 POKEV1,17:POKEH1,15:POKEVO,15
1060 FORJ=1TO10
1070 FORK=1TO50:POKEL1,K:NEXTK
1080 FORK=50TO1STEP-1:POKEL1,K:NEXTK,J
1090 POKEVO,0:POKEV1,0:POKEH1,0:POKEL1,0
1100 POKE53281,6
1110 BO=BO-1
1120 RETURN
1130 REM INSTRUCTIONS
1140 PRINT"{SC}"
1150 PRINTSPC(15);"{BK}SPACE GUARD"
1160 PRINTSPC(10);"{CD}{CD}{CY}WITH YOUR SHIP {YL}=>{CY}"
1170 PRINTSPC(10);"{CD}YOU ARE TO COLLECT"
1180 PRINTSPC(10);"{CD}THE SATELITES IN EACH"
1190 PRINTSPC(10);"{CD}SECTOR => {BK}* ";CHR$(218);" # ";CHR
$(209);"{CY}"
1200 PRINTSPC(10);"{CD}BE CAREFUL TO AVOID"
1210 PRINTSPC(10);"{CD}THE EVIL ALIEN SHIPS:"
1220 PRINTSPC(10);"{CD}{CD}{YL}=>"
1230 PRINTSPC(10);"{CD}{CD}=>";SPC(12);"=>{CY}"
1240 POKE53248,245:POKE53249,72:POKE53250,150:POKE53251,176
1250 POKE53252,150:POKE53253,204:POKE53254,245:POKE53255,204
1260 POKE53269,15
1270 POKE53287,4:POKE53288,7:POKE53289,3:POKE53290,5
1280 POKE252,2:POKE253,1
1290 FORI=1TO110:SYS16998:FORJ=1TO10:NEXTJ,I
1300 POKE252,6:POKE253,3
1310 FORI=1TO40:SYS16896:FORJ=1TO10:NEXTJ,I
1320 POKE252,4:POKE253,2
1330 FORI=1TO58:SYS16998:FORJ=1TO10:NEXTJ,I
1340 POKE252,2:POKE253,1
1350 FORI=1TO80:SYS16896:FORJ=1TO10:NEXTJ,I
1360 POKE252,4:POKE253,2
```



```
1370 FORI=1TO50:SYS16896:FORJ=1TO10:NEXTJ,I
1380 PRINTSPC(15);"{YL}{CD}PRESS A KEY"
1390 GOSUB170
1400 POKE53264,0:POKE53269,0
1410 PRINT"{SC}"
1420 PRINTSPC(15);"{WH}SPACE GUARD"
1430 PRINTSPC(8);"{CD}{CD}{CY}USE A JOYSTICK FOR CONTROL."
1440 PRINTSPC(8);"{CD}YOU CAN MAKE FIVE"
1450 PRINTSPC(8);"{CD}HYPERSPACE JUMPS WITH"
1460 PRINTSPC(8);"{CD}EACH OF YOUR FIVE SHIPS."
1470 PRINTSPC(8);"{CD}MORE HYPERSPACE JUMPS ARE"
1480 PRINTSPC(8);"{CD}AWARDED TO SUPERIOR PLAYERS."
1490 PRINTSPC(8);"{CD}BONUS SHIPS EVERY 10000"
1500 PRINTSPC(15);"{YL}{CD}PRESS A KEY"
1510 GOSUB170
1520 RETURN
1530 REMNEW HIGH SCORE
1540 HS=PP:HS$=""
1550 PRINT"{SC}{CD}{CD}{CD}"
1560 PRINTSPC(15);"{RV}{WH}A NEW HIGH!{RO}"
1570 PRINTSPC(10);"{CD}{BK}ENTER YOUR INITIALS"
1580 PRINTSPC(18);"{CD}---{CL}{CL}{CL}";
1590 FORI=1TO3
1600 GOSUB170
1610 HS$=HS$+RE$
1620 PRINTRE$;
1630 NEXTI
1640 RETURN
1650 P=255*(PEEK(53264)AND1)
1660 P=INT((PEEK(53248)-23+P)/8)+INT((PEEK(53249)-20)/8)*40+
905
1670 IFP<2023ANDP>1023ANDPEEK(P)<>32THEN1700
1680 POKEVO,0:POKEH1,0:POKEV1,0
1690 RETURN
1700 GOSUB280
1710 POKEP,32:S=S+1
1720 POKEVO,0:POKEH1,0:POKEV1,0
1730 RETURN
1740 REM SPRITE EXPLOSION
1750 D1=PEEK(2040):D2=PEEK(53287)
1760 POKEH1,10:POKEV1,129:POKEVO,15
1770 MN=MN-1
1780 FORI=0TO2
1790 POKE2040,208+I
1800 FORJ=1TO9:POKE53287,EXC(J):FORK=1TO25:NEXTK:POKEH1,PEEK
(L1)+J:NEXTJ
```



```
1810 NEXTI
1820 POKE53269,PEEK(53269)AND(NOT1)
1830 POKE2040,D1:POKE53287,D2
1840 POKE53278,0:POKE53264,PEEK(53264)AND254
1850 POKEH1,20
1860 POKE53248,100:POKE53249,100
1870 FORI=15TO0STEP-1
1880 POKEVO,I
1890 FORJ=1TO30:NEXTJ,I
1900 POKEH1,0:POKEV1,0:POKEVO,0
1910 RETURN
1920 REM MOVE NME SPRITES
1930 N=L/6:IFN>3THENN=3
1940 FORI=1TON+1
1950 SYS17189
1960 NEXTI
1970 RETURN
1980 REM UPDATE PLAYER MOVEMENT
1990 N=L/6:IFN>3THENN=3
2000 FORI=1TON+1
2010 SYS16384
2020 NEXTI
2030 RETURN
2040 REM INCREASE SPRITE SIZE BY LEVEL
2050 POKESH,2:RETURN
2060 POKESH,4:RETURN
2070 SL=L:RETURN
2080 POKESH,8:RETURN
2090 POKESH,10:RETURN
2100 SL=L+1:RETURN
2110 POKESH,14:RETURN
2120 POKESV,2:RETURN
2130 SL=L+2:RETURN
2140 POKESV,4:RETURN
2150 POKESV,8:RETURN
2160 POKESV,10:RETURN
2170 POKESV,14:RETURN
2180 REM HYPERSPACE
2190 IFHY=0THENRETURN
2200 HY=HY-1
2210 POKE53269,PEEK(53269)AND254
2220 GOSUB310
2230 POKE53264,PEEK(53264)AND254
2240 POKE53248,FNRAN(240)+8:POKE53249,FNRAN(206)+32
```

```
2250 POKE53269,PEEK(53269)OR1
2260 RETURN
2270 POKE53264,PEEK(53264)OR1
2280 POKE53248,FNRAN(72):POKE53249,FNRAN(206)+32
2290 POKE53269,PEEK(53269)OR1
2300 RETURN
2310 REM MAINLINE
2320 REM SETUP EXPLOSION COLOURS
2330 FORI=1TO9:READEXC(I):NEXTI
2340 REM SETUP SPRITES
2350 FORI=0TO6
2360 READA
2370 FORJ=0TO62:READB:POKEA+J,B:NEXTJ
2380 NEXTI
2390 POKE2040,11:POKE2041,13:POKE2042,14:POKE2043,15
2400 FORI=53248TO53255
2410 POKEI,0
2420 NEXTI
2430 REM CLEAR SPRITE REGISTERS
2440 POKE53278,0:POKE53279,0
2450 POKE53271,0:POKE53277,0
2460 POKE53269,0:POKE53264,0
2470 REM SET LOGICAL AND VALUES
2480 POKE829,2:POKE831,4:POKE833,8
2490 REM SETUP RANDOM FUNCTION
2500 DEF FN RAN(X)=INT(RND(1)*X+1)
2510 SV=53271:SH=53277
2520 CO=55296:SC=1024
2530 HY=5:S=-1:MN=5
2540 P1=0:PP=0:BO=0
2550 FORI=54272TO54296:POKEI,0:NEXTI
2560 POKE54277,64:POKE54278,64
2570 VO=54296:V1=54276:H1=54273:L1=54272
2580 IFHS=0THENGOSUB1130
2590 PRINT"{SC}{CD}{CD}{CD}"
2600 PRINTSPC(8);"CHOOSE A NUMBER FROM 1-9"
2610 MIN=1:MAX=9:GOSUB210
2620 L=RE*10-9
2630 PRINTSPC(12);"{CD}{CD}PRESS A KEY TO START"
2640 GOSUB170
2650 SL=L-1
2660 IFL-3<1THEN2720
2670 FORI=1TOL-3
2680 ONIGOSUB2050,2060,2070,2080,2090,2100
```

```
2690 NEXTI
2700 REM SET UP SCREEN
2710 IFS=>0THENGOSUB960
2720 POKE53269,0
2730 IFMN=0THEN3010
2740 BO=BO+1:S=0
2750 IFBO>4THENBO=1:L=L+1:HY=HY+1
2760 IFBO<1THENBO=1
2770 IFL>255THENL=255
2780 PRINT"{SC}"
2790 ONBOGOSUB390,450,510,570
2800 REM GAME LOOP
2810 POKE53287,4:POKE53288,7:POKE53289,3:POKE53290,5
2820 POKE53248,50:POKE53249,50
2830 POKE53250,20:POKE53251,200
2840 POKE53252,250:POKE53253,50
2850 POKE53254,20:POKE53255,200
2860 POKE53264,8:POKE53269,15:POKE53278,0
2870 IFL-3<0THEN2910
2880 ONL-3GOSUB2050,2060,2070,2080,2090,2100,2110,2120,2130
2890 IFL-11<0THEN2910
2900 ONL-11GOSUB2140,2150,2160,2170
2910 IFL>20THENSL=SL+3
2920 SP=FNRAN(3):POKE252,2*SP:POKE253,SP
2930 GOSUB1930
2940 GOSUB1990
2950 IF(PEEK(53278)AND1)THENGOSUB1740:POKE53269,PEEK(53269)O
R1:GOTO2710
2960 IFS=10THENGOSUB640:GOTO2720
2970 ON-((PEEK(56320)AND16)=0)GOSUB2190
2980 ON(PEEK(53279)AND1)GOSUB1650
2990 GOTO2920
3000 REM END OF GAME
3010 IFPP>HSTHENGOSUB1540
3020 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}"
3030 PRINTSPC(16);"{WH}GAME OVER"
3040 PRINTSPC(15);"{CD}{CD}{YL}FINAL SCORE{CD}{CD}"
3050 PRINTSPC(18);PP
3060 PRINTSPC(9);"{CD}{GR}";HS$;" HAS A HIGH SCORE OF";HS
3070 PRINTSPC(9);"{CD}{CD}{CY}PRESS {RV}P{RO} TO PLAY AGAIN"
3080 PRINTSPC(7);"{CD}PRESS ANYTHING ELSE TO END"
3090 GOSUB170
3100 IFRE$="P"THEN2430
3110 PRINT"{SC}"
3120 END
```

```
3130 REM EXPLOSION COLOURS
3140 DATA0,11,12,15,1,7,8,10,2
3150 REM SPRITES
3160 DATA704
3170 DATA0,8,0,0,8,0,0,8,0,0,28,0,0,28,0,8,62,8,4,65,16
3180 DATA3,128,224,1,0,64,1,0,64,9,0,112,1,0,64,1,0,64
3190 DATA3,128,224,4,65,16,8,62,8,0,28,0,0,28,0,0,8,0,0,8,0,
0,8,0
3200 DATA832
3210 DATA0,65,0,0,162,128,1,26,64,2,0,32,4,0,16,8,0,8,16,0,4
3220 DATA32,0,2,255,255,255,34,34,34,34,34,34,34,34,34,255,2
55,255
3230 DATA32,0,2,16,0,4,8,0,8,4,0,16,2,0,32,1,26,64,0,162,128
,0,65,0
3240 DATA896
3250 DATA20,0,40,26,0,88,25,0,152,248,0,31,120,0,30,133,0,16
1,67,129,194
3260 DATA63,0,252,2,189,64,0,66,0,3,143,194,0,66,0,2,189,64,
63,0,252
3270 DATA67,129,194,133,0,161,120,0,30,248,0,31,25,0,152,26,
0,88,20,0,40
3280 DATA960
3290 DATA0,0,0,0,0,0,4,0,32,4,0,32,2,0,64,2,0,64,1,60,128,1,
66,128,0,129,0
3300 DATA0,129,0,1,0,128,1,36,128,2,36,64,2,36,64,4,153,32,8
,0,16
3310 DATA31,60,248,56,195,28,48,0,12,64,0,2,128,0,1
3320 DATA13312
3330 DATA48,3,131,0,132,80,71,88,42,4,32,42,14,3,16,16,25,0,
38,1,196
3340 DATA32,145,0,144,241,0,145,241,194,52,251,8,34,2,4,17,2
52,4
3350 DATA16,124,24,48,24,0,12,0,80,66,6,68,1,73,16,32,176,2,
48,0,231,0,3,0
3360 DATA13376
3370 DATA128,4,1,0,0,0,32,16,0,0,0,8,0,8,16,4,0,0,4,0,0,0,16
,128
3380 DATA0,144,0,0,130,6,12,4,0,160,32,32,0,129,0,0,8,128,0,
8,32
3390 DATA2,8,32,0,0,0,16,16,8,32,0,0,0,0,0,128,16,1
3400 DATA13440
3410 DATA0,0,0,0,0,0,0,16,0,0,0,0,0,0,16,4,0,0,0,0,0,0,0,128
3420 DATA0,128,0,0,130,0,0,0,0,32,0,32,0,0,0,0,0,0,2,8,64
3430 DATA0,0,0,0,0,0,0,0,0,8,0,0,0,0,0,0,8,0
```

HOW A GAME IS MADE

Every game starts in the same place... in someone's head. The idea is then put down on paper. All the features the game will have are written down. Pictures of the various characters and backgrounds are drawn. Every rule and aspect of the game is included in this paper plan.

The next step is to put all this information into an order of events. On another piece of paper shapes are drawn and each event of the game is put in a box, circle or diamond. Then, with each figure a brief note of purpose is made. The name of this series of shapes and notes is a flowchart.

Every event in the flowchart is a small program in itself. These small programs are commonly called subroutines. Breaking all the events into subroutines makes the task of programming the game much easier. Tracing a flaw in a particular subroutine is easier than tracking one down in a long, unco-ordinated program.

Quite often subroutines can be used more than once. They can even be transferred from one game to another. There is no point in designing a new subroutine to examine which way a player has moved the joystick for every game using a joystick. By using some of the same subroutines from game to game, a programmer will save himself or herself a lot of time.

Here are some guidelines to follow when designing a game:

1. Write down your ideas.	2. Draw up a flowchart.
3. Use subroutines.	4. Use the same subroutines where possible.

The game METEOR SHOWER started with these ideas:

— the player will be able to move the ship in a bounded area on the screen

— the ship will rotate on its axis

— the ship will be able to move forward to fire its laser in the direction in which it is pointing

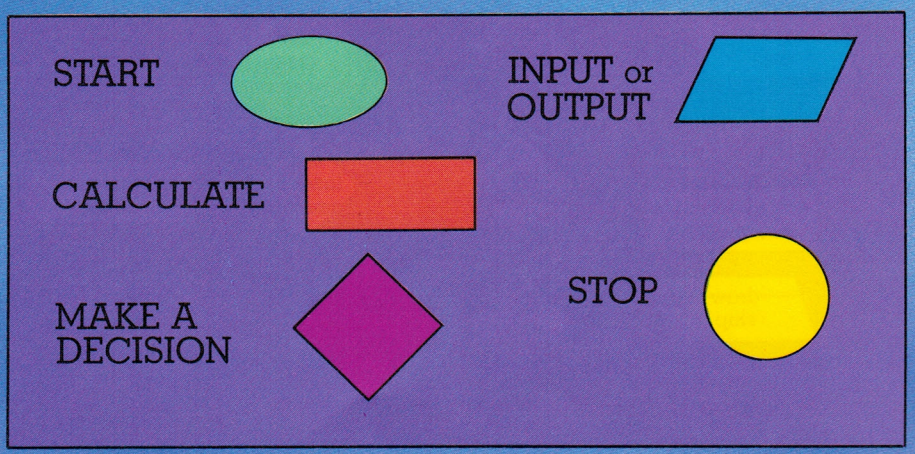
— to challenge the player, asteroids will randomly appear and move across the screen

— points will be awarded for the laser destruction of asteroids

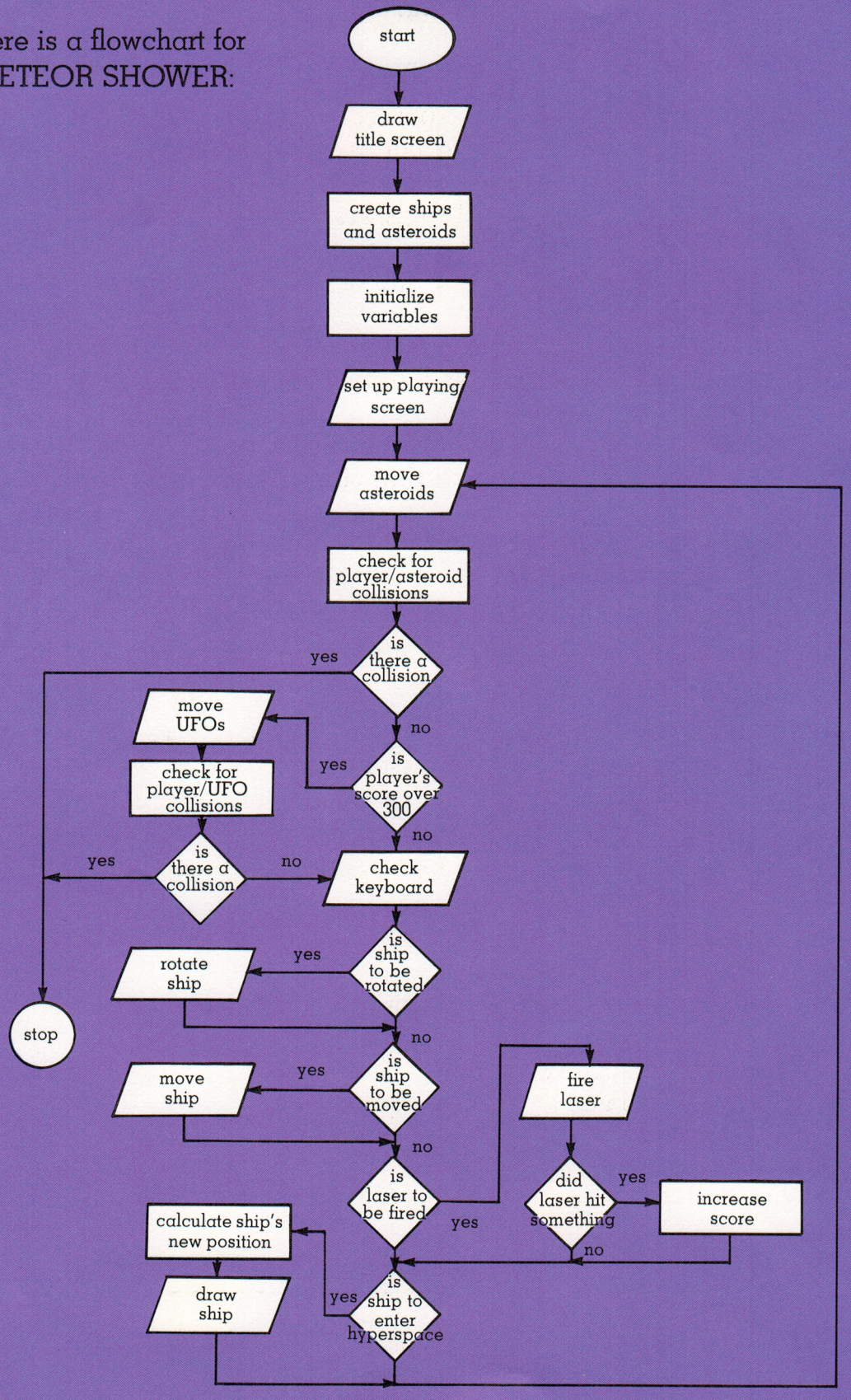
— the player may have the ship enter hyperspace (it will disappear and reappear elsewhere on the screen), but he will lose points for this

— when the player has reached a certain score, alien ships (UFOs) will also appear on the screen to menace him

These ideas were then summarized in a flowchart. The shapes in the flowchart mean:



Here is a flowchart for METEOR SHOWER:



Here is a verbal description of how the game METEOR SHOWER works:

Line numbers	Operation
100 - 130	move character set from ROM
140 - 300	data statements store values for sound making and character set redesigning
310 - 350	clear and set sound registers
340	sets top of BASIC memory
350 - 420	changes screen color and prints title screen
430 - 450	plays sound
460 - 470	waits for a key to be pressed
480	creates spaceship and meteor shapes
520	sets up direction data
530	turns on the special characters
540 - 560	sets score to zero and puts ship in starting position, sets display colors
570 - 580	sets up screen boundaries
590 - 900	main game loop
610 - 770	checks asteroid positions for possible collisions; if player's score is over 300, starts putting UFOs on screen; checks UFO positions
780	turns sound off
790	checks the keyboard if asteroid path is not empty
800	draws asteroids
820 - 900	checks keyboard
880	prints score
890	draws ship
900	end of game loop
910 - 1040	subroutine to move ship
1030 - 1080	fires laser
1090 - 1100	increases score if asteroid hit
1110	prints score on screen
1120 - 1140	clears laser shot from screen
1150 - 1250	end of game
1260 - 1320	play again option

By examining the ideas, the flowchart, the listing description and the program listing itself, you will be able to see how METEOR SHOWER fits together. You may follow these same steps when creating games of your own. **HAVE FUN!**

More
GETTING AHEAD
Computer Books:

FANTASTIC GAMES
for the VIC 20
Six fantastic games to key into your own VIC 20 computer!

FANTASTIC GAMES
for the Commodore 64
Six arcade style games. Challenging fun for the novice and expert alike!

APPLE GRAPHICS
Learn to program your own graphics games on the APPLE computer starting with the great space game, Appleflight!

EXPLORING THE VIC
From hook-up to peripherals, programming to expansion memory – the VIC 20 explained! Includes a terrific game!

HAYES PUBLISHING LTD.
3312 MAINWAY
BURLINGTON, ONTARIO
L7M 1A7

ISBN 0-88625-066-8

Printed in Hong Kong